



**UNIVERSIDADE DO MINDELO**  
**DEPARTAMENTO DE ENGENHARIA E**  
**RECURSOS DO MAR**

**CURSO DE LICENCIATURA EM INFORMÁTICA DE GESTÃO**

**RELATÓRIO DE PROJETO DE LICENCIATURA**

Ano letivo 2016/2017 – 4º Ano

**Autor: Yanik Gilson Santos Oliveira, N.º 2858**

**Mindeló, Julho 2017**

**Yanik Gilson Santos Oliveira**

**QSMOBILE: *APP* e Plataforma Web para Fila  
de Espera**

**Universidade do Mindelo**

**Mindelo, Julho de 2017**





**UNIVERSIDADE DO MINDELO**  
**DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR**

# **CURSO DE LICENCIATURA EM INFORMÁTICA DE GESTÃO**

**RELATÓRIO DE PROJETO DE LICENCIATURA**  
**Ano letivo 2016/2017 – 4º Ano**

**Autor: Yanik Gilson Santos Oliveira, N.º 2858**

**Mindeló, Julho de 2017**

**Yanik Gilson Santos Oliveira**

**QSMOBILE: *APP* e Plataforma Web para Fila  
de Espera**

---

Trabalho apresentado à Universidade do Mindelo  
como parte dos requisitos para obtenção do grau de  
Licenciatura em Informática de Gestão

Orientador: Mestre Paulo Santos Silva

## RESUMO

Nos dias de hoje tecnologias de diversos tipos tem vindo a ganhar cada vez mais espaço na nossa sociedade. Há cada vez mais procura por novas tecnologias que visam facilitar cada vez mais a vida das pessoas.

O presente trabalho tem como objetivo o desenvolvimento e apresentação de uma aplicação android para filas de espera denominada de QSmobile (Sistema De Fila Móvel) e o seu Web Site. Essa aplicação permite ao utilizador consultar filas de espera de várias empresas ou instituições que possuem os sistemas de gestão de filas de esperas através do seu telemóvel, desde que esse tenha acesso a internet e também desde que a empresas ou instituição esteja disponível na aplicação. Ao retirar uma senha e estiver faltando muito para ser atendido o utilizador poderá se deslocar a qualquer lugar que pretenda, ligar e internet e aceder a aplicação para controlar dali a sua vez na fila. Poderá também ativar o sistema de notificações e receber momentaneamente notificações de alarme que permitam com que o utilizador não esqueça de consultar o seu telemóvel e perder a sua vez. A aplicação foi feita para a plataforma Android e utiliza uma Webservice em PHP para ter acesso aos dados das filas na base de dados no MySQL.

The Web Site's main objective is to enable companies to contact to order the service and be inserted in the App, so that customers can see your queue. The WebSite was made with HTML, CSS, PHP, and uses a database made with MYSQL.

**Palavras-chave:** Android, Computação Móvel, Web Service, SQLite, Filas de Espera, JSON.

## **ABSTRACT**

Nowadays technologies of various types have been gaining more and more space in our society. There is an increasing demand for new technologies to make life easier for people.

The present work aims at the development and presentation of an android application to queues named QSmobile (queue system mobile) and your Web Site. This application allows the user to query for queues of several companies or institutions that have queue management systems through your phone waits, provided he has access to the internet and also since the companies or institutions are available on application. To remove a password and is missing much to be granted the user can travel anywhere you want, call and internet and access application to control in your time in the queue. You can also activate the notification system and receive alarm notifications momentarily allowing the user don't forget to consult your cell phone and lose your time. The application was made to the Android platform and use a WebService in PHP to access the data from the rows in the database in MySQL.

The Website's main objective is to enable companies to contact to order the service and be inserted in the App, so that customers can see your queue. The Web Site was made with HTML, CSS, PHP, and uses a database made with MYSQL.

**Keywords:** Android, Mobile Computing, Web Service, SQLite, Queues, JSON.

## **DEDICATORIA**

Dedico esse trabalho a todos que acreditaram em mi e me apoiaram ao longo desse grande percurso, principalmente a minha família, professores, colegas e amigos.



## **AGRADECIMENTOS**

Durante a realização desse trabalho científico tive muito apoio por parte de varias pessoas, por isso é com muita satisfação que agradeço a todos que contribuíram de alguma forma na realização desse trabalho.

Agradeço a Deus, por estar presente em todos os momentos da minha vida, me dando sempre força e muita coragem perante todos os obstáculos.

Aos meus pais João Oliveira e Marcelina Dos Santos e aos meus irmãos Lonardy Oliveira e Enia Oliveira por todo o apoio, amor e carinho que me foi transmitido todos esses anos.

Ao meu orientador Mestre Paulo Santos Silva por acolher este meu projeto e por disponibilizar o seu tempo e conhecimento ao longo desse período. Sem si ficaria muito mais difícil conseguir concluir esse objetivo.

A todos os docentes do curso pelas palavras de incentivo e conselhos dados ao longo desse percurso.

Por fim, um agradecimento sincero a todos os meus colegas de turma que de uma forma ou de outra contribuíram para a realização deste trabalho.

A todos, o meu mais sincero obrigado!

*“Algo só é impossível até que alguém duvide e resolva  
provar ao contrário.” – **Albert Einstein***

# Sumário

<b>CAPÍTULO I .....</b>	<b>1</b>
<b>1. Introdução .....</b>	<b>1</b>
1.1. Contextualização.....	1
1.2. Motivação .....	2
1.3. Objetivos .....	3
1.3.1. Geral.....	3
1.3.2. Específicos .....	3
1.4. Metodologia .....	4
1.5. Estrutura do trabalho.....	5
<b>CAPÍTULO II.....</b>	<b>6</b>
<b>2. Enquadramento .....</b>	<b>6</b>
2.1. Filas de espera.....	7
2.1.1. Sistemas de gestão de filas de espera.....	7
2.1.1.1. Componentes de um sistema de Filas de Espera .....	8
2.1.1.2. Tipos de Sistemas existentes.....	9
2.2. Android .....	17
2.2.1. Estrutura Geral da plataforma Google Android.....	18
2.2.1.1. Arquitetura do android.....	18
2.2.1.2. Aplicações.....	19
2.2.1.3. Bibliotecas .....	19
2.2.1.4. Android Runtime .....	20
2.2.1.5. Linux Kernel .....	21
2.2.2. Componentes de uma aplicação.....	21
2.2.3. Ciclo de vida de uma aplicação .....	23
2.2.4. A execução das aplicações.....	23
2.2.5. Versões recentes .....	25
2.2.5.1. Android 6.0 Marshmallow .....	25
2.2.5.2. Android Nougat (7.0 e 7.1) .....	25
<b>CAPÍTULO III .....</b>	<b>27</b>
<b>3. Tecnologias Utilizadas.....</b>	<b>27</b>

3.1.	Linguagem JAVA .....	27
3.2.	Android Estúdio .....	28
3.2.1.	Android SDK .....	29
3.3.	Linguagem PHP .....	30
3.4.	Base de Dados MySQL.....	31
3.5.	Base de dados SQLite .....	32
3.6.	WebService RESTful.....	33
3.7.	Json .....	34
3.8.	Genymotion .....	35
3.9.	Dream Weaver .....	37
3.10.	Linguagem HTML .....	39
3.11.	Linguagem CSS .....	40
<b>CAPÍTULO IV .....</b>		<b>42</b>
<b>4.</b>	<b>Análise do sistema a desenvolver .....</b>	<b>42</b>
4.1.	Visão geral do Sistema .....	42
4.2.	Análises de Requisitos do Sistema .....	43
4.2.1.	Requisitos funcionais .....	43
4.2.2.	Requisitos não funcionais .....	45
4.3.	Desenho do Sistema.....	46
4.3.1.	Diagramas de Caso de Uso .....	47
4.3.2.	Diagrama de Atividade .....	48
4.3.3.	Diagrama de Classes .....	49
4.3.4.	Diagrama de Sequência .....	50
4.3.5.	Base de dados.....	52
<b>CAPÍTULO V .....</b>		<b>54</b>
<b>5.</b>	<b>Implementação do Protótipo .....</b>	<b>54</b>
5.1.	Configuração da aplicação .....	54
5.2.	Criação da base de dados local (SQLite) .....	56
5.3.	Acessando o Webservice .....	56
5.4.	Obter dados das filas.....	57
5.5.	Service (Notification Manager) .....	57
5.6.	Envio de Email (comunicar problemas) .....	58

5.7.	Adaptação da Website para dispositivos móveis .....	59
5.8.	Funcionamento do projeto .....	60
5.8.1.	Aplicação qsmobile.....	60
5.8.2.	website .....	67
<b>CAPÍTULO VI .....</b>		<b>72</b>
<b>6.</b>	<b>Conclusão .....</b>	<b>72</b>
6.1.	Resultados obtidos .....	72
6.2.	Trabalhos futuros .....	72
<b>Referências Bibliográficas .....</b>		<b>74</b>
<b>Anexos.....</b>		<b>78</b>
A)	Outras partes da Website .....	78
B)	Simulador de um sistema de gestão de filas de espera .....	79
C)	Termo de Aceitação .....	81

## LISTA DE FIGURAS

Figura 1-Componentes de um sistema de Filas de Espera .....	8
Figura 2- Componentes da QSystems .....	11
Figura 3-Qmagine.....	12
Figura 4-Funcionamento do sistema INLINE .....	14
Figura 5- Exemplo de display da INLINE .....	15
Figura 6- Funcionamento da Q.Track .....	17
Figura 7-Arquitetura plataforma Android .....	19
Figura 8-Ciclo de vida de uma Activity .....	23
Figura 9- Processo de geração de aplicações.....	24
Figura 10-Exemplo de estrutura Json .....	34
Figura 11-Exemplo de Estrutura XML.....	35
Figura 12-Arquitetura do sistema proposto .....	43
Figura 13-Diagrama de Use Case App Qsmobile .....	48
Figura 14- Diagrama de use case para utilizadores da Website .....	48
Figura 15-Diagrama de Atividades da App QSmobile.....	49
Figura 16-Diagrama de classe da App android .....	50
Figura 17-Diagrama de sequência para caso de uso requisitar dados das filas .....	51
Figura 18-Diagrama de sequência de caso de uso solicitar alertas por notificação .....	51
Figura 19-Tabelas da base de dados da Website .....	52
Figura 20-Tabela da base de dados da aplicação android .....	53
Figura 21- Configuração do arquivo AndroidManifest.xml.....	55
Figura 22-Classe DataBaseHelper .....	56
Figura 23- Constituição da classe LoginDataBaseHelper .....	56
Figura 24- Método para fazer leitura e disponibilização dos dados .....	57
Figura 25-Dados de uma fila de espera .....	57
Figura 26- Código para disparar notificações.....	58
Figura 27- Envio de emails.....	59
Figura 28-Adaptação para dispositivos móveis 980px .....	60
Figura 29-Formulário para cadastro do utilizador .....	61
Figura 30-Formulario de login .....	61
Figura 31-Menu da aplicação .....	62
Figura 32-Lista de empresas disponíveis .....	63
Figura 33-Layout de Fila de espera Ex: BCA .....	64
Figura 34-Painel de notificações .....	65
Figura 35- Exemplo de notificação entrando .....	65
Figura 36-Conectando com cliente.....	66
Figura 37-Composição de um email.....	66
Figura 38- Tela de ajuda para utilizadores .....	67
Figura 39-Parte inicial para público .....	68
Figura 40- Cadastro de empresas .....	68
Figura 41-Formulario de contato para requisitar o serviço .....	69
Figura 42-Pagina inicial do Website num dispositivo móvel.....	69
Figura 43- Menu da Website no android .....	70
Figura 44-Painel do administrador .....	71
Figura 45- Confirmação dos pedidos através de email.....	71

## LISTA DE SIGLAS E ABREVIATURAS

<b>TCP</b>	<i>Transmission Control Protocol</i>
<b>URL</b>	<i>Uniform Resource Location</i>
<b>UML</b>	<i>Unified Modeling Language</i>
<b>PHP</b>	<i>Personal Home Page</i>
<b>HTTP</b>	<i>HyperText Transfer Protocol</i>
<b>HTML</b>	<i>Hypertext Markup Language</i>
<b>WEB</b>	<i>World Wide Web</i>
<b>ADT</b>	<i>Android Development Tools</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>OS</b>	<i>Operational System</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>XML</b>	<i>Extensible Markup Language</i>
<b>SQL</b>	<i>Structured Query Language</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>SGBD</b>	<i>Sistema de Gerenciamento de Banco de Dados</i>
<b>REST</b>	<i>Representation State Transfer</i>
<b>IDC</b>	<i>International Data Corporation</i>
<b>PC's</b>	<i>Personal Computers</i>
<b>CGI</b>	<i>Common Gateway Interface</i>
<b>VM</b>	<i>Virtual Machine</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>APK</b>	<i>Android Package Kit</i>

# CAPÍTULO I

## 1. Introdução

### 1.1. Contextualização

Nos dias de hoje, ninguém consegue ficar longe de um telemóvel, seja para mandar um email, tirar uma foto, assistir um vídeo, conversar com os amigos, navegar na internet, acompanhar as redes sociais entre outras varias coisas que podem fazer. Portanto, os smartphones ou tablets atualmente são objetos praticamente inseparáveis da maioria das pessoas.

Segundo pesquisas, mais de 3 bilhões de pessoas têm um telefone telemóvel, e o mercado de aplicações móveis transformou-se numa área muito viável, rendendo bilhões todos os anos. E há que fazer uma análise de duas formas. Os utilizadores comuns atualmente buscam um telemóvel com um visual sofisticado, moderno, de fácil navegação, assim como uma infinidade de aplicações e novos recursos. Tanto as empresas quanto os desenvolvedores buscam uma plataforma moderna e ágil para o desenvolvimento das suas aplicações. Os fabricantes como a LG, Motorola, Samsung, HTC, Intel, Sony e algumas outra, precisam de uma plataforma robusta e com muitas funcionalidades para lançar no mercado os seus produtos. E é aqui que o Android entra, pois ele é perfeito para todos os casos.

Feitas algumas pesquisas notei que hoje o Android está disponível em muitas plataformas, como smartphones, tablets, TV (Google TV), relógios (Android Wear), óculos (Google Glass), carros (Android Auto) e é o sistema operacional móvel mais utilizado no mundo.

Há que ser mencionado que o mercado corporativo também está cada vez mais utilizando esses recursos móveis, tanto que diversas empresas estão buscando incorporar aplicações móveis a seu dia a dia para agilizar seus negócios. As empresas obviamente visam o lucro, por isso, tanto os smartphones quanto os tablets ocupam um importante espaço em um mundo em que a palavra “mobilidade” está se tornando cada vez mais conhecida.



Dentro desse contexto, estamos diante de uma excelente oportunidade, pois o mobile é um grande pilar na área de tecnologia e segundo pesquisas é uma das áreas que mais vai crescer nos próximos anos.

Tendo em conta esses vários aspetos referidos acerca do android e a atualidade vi que seria muito interessante desenvolver uma aplicação android com um tema viável que procura-se facilitar a vida das pessoas e dar algum contributo a sociedade.

Tenho vindo a reparar que muitas empresas atualmente utilizam os sistemas de gestão de filas de espera por causa do grande número de clientes que possuem. Tendo em conta algumas limitações demonstradas por essa tecnologia seria algo inovador estender as funcionalidades desses sistemas integrando-os agora na plataforma android com uma aplicação para filas de espera que permite-se ao utilizador ou cliente consultar a fila de espera mesmo estando fora do estabelecimento.

## **1.2. Motivação**

O primeiro fator que me motivou na implementação desse projeto foi o fato de que atualmente tem sido cada vez mais difícil chegar num estabelecimento e ser atendido com rapidez, já que muitos deles possuem um elevado número de clientela. Mesmo com os sistemas de gestão filas de espera implementadas em várias empresas e instituições muitas pessoas tem gasto muito tempo nas filas esperando ser atendido, o que me levou a constatar que esses sistemas não tem conseguido ajudar a reduzir significativamente o tempo de espera nas filas mas sim só a organizar os clientes nas filas. Tenho visto muito descontentamento por parte da população quanto a esse tema o que é compreensível, visto que muitas pessoas perdem tempo nas filas quando por exemplo tem que ir trabalhar ou fazer qualquer outra coisa.

Quando há muitas pessoas na fila as vezes o cliente acaba por ir fazer alguma outra coisa noutro lugar para assim ganhar algum tempo, mas quando volta as vezes depara com a situação de ter perdido a sua vez, agora terá que voltar a retirar uma nova senha visto que a maioria das empresas como a caixa económica por exemplo dão tolerância de até 3 senhas depois de chamar o seu número, apos isso se o cliente

ainda não tiver comparecido perderá a sua vez causando todo aquele constrangimento.

Outro fator que me motivou foi o grande aumento de utilização de dispositivos móveis não só em cabo verde mas também no mundo todo e por causa disso algumas empresas em Cabo Verde tem-se dedicado muito na venda de dispositivos móveis com diversos sistemas operativos, agora tornou-se mais fácil alguém conseguir arranjar um telemóvel.

Depois desse aumento as atenções tem voltado para programação de aplicações móveis que tem vindo a crescer de uma forma bem interessante no mercado, mostrando ser muito viável e quem sabe no futuro uma boa oportunidade para mim caso eu queira seguir nesse ramo de tecnologias.

Tendo em conta esses pontos surgiu-me a ideia desse projeto, e em conversas com o meu orientador vi que seria algo inovador e com algum futuro implementar uma aplicação móvel que permite-se ao utilizador controlar a sua vez numa fila de espera mesmo a distancia ou seja fora do estabelecimento, isso provavelmente diminuiria o seu tempo de espera nas filas e diminuiria também toda aquela multidão dentro do estabelecimento esperando ser atendido e assim evitaria muitos constrangimentos.

### **1.3. Objetivos**

#### **1.3.1. GERAL**

O objetivo geral desse projeto é desenvolver uma aplicação android que permita ao utilizador controlar a sua vez nas filas de espera de qualquer empresa ou instituição que estejam disponíveis na aplicação desde que se tenha acesso a internet. E também um Website que permita disponibilizar toda a informação acerca da aplicação e também permitir às empresas entrarem em contato comigo para solicitar o serviço e serem inseridas na APP.

#### **1.3.2. ESPECÍFICOS**

##### **App Android:**

- Implementar o sistema utilizando a plataforma Android;

- Criar sistema de Cadastro e login com Base de Dados SQLite;
- Mostrar informações das empresas disponíveis;
- Comunicar com a base de dados Mysql através de um Webservice desenvolvido em *PHP*;
- Permitir acesso aos Sites das empresas disponíveis na APP;
- Utilizar Service para disparar notificações mesmo com a App fechada;
- Utilizar ACTION\_SEND e emailIntent para envio de Email com (Outlook, Gmail ou qualquer outro);

#### **WebSite:**

- Criar o Website utilizando HTML, CSS, PHP;
- Utilizar banco de dados *MySQL*;
- Permitir cadastro e login de empresas;
- Mostrar todas as informações da APP;
- Formulário de contato para requisição do serviço da APP;
- Permitir download da APP;
- Criar a parte do administrador para gerir os utilizadores e os pedidos (confirmar com envio de email).
- Adaptar para dispositivos móveis;

### **1.4. Metodologia**

Numa primeira fase foram feitas varias pesquisas e estudos dentro do contexto do meu projeto tanto em bibliografias tanto na internet. Na fase seguinte foi feita uma análise do projeto ou seja foram levantadas todos os requisitos do mesmo e procurando tudo o que seria importante estar nesse protótipo. Na terceira fase foi feito a modelação completa do protótipo.

Depois seguiu-se para a quarta fase onde foi instalado todo o ambiente de desenvolvimento. Logo na quinta fase foi desenvolvido todo o protótipo com todas as funcionalidades previstas inicialmente. Na sexta fase já com o sistema todo implementado foram feitos os testes para avaliação das funcionalidades do protótipo. E para concluir, na sétima fase foi feito o relatório.

## **1.5. Estrutura do trabalho**

Tendo em conta todos os objetivos que foram definidos para esse projeto este relatório está dividido em 6 capítulos.

Logo no início temos a parte de introdução que contem a contextualização, motivação, objetivos e metodologia. No segundo capítulo fez se uma abordagem teórica de alguns dos sistemas de gestão de filas existentes e também acerca da plataforma android. No terceiro capítulo descrevi as tecnologias e ferramentas utilizadas para implementação desse projeto. No capítulo quatro fez-se uma análise de todo o sistema a desenvolver, nessa parte se encontra a arquitetura geral do sistema, a sua modelação e todos os requisitos previstos. No quinto capítulo temos a parte da implementação onde mostro o funcionamento de todo o sistema, apresentando algumas das telas principais e alguns trechos de códigos. O sexto capítulo trata-se das considerações finais onde descrevo os resultados obtidos e os trabalhos que pretendo implementar futuramente.

### 2. Enquadramento

Atualmente estamos vivendo num mundo muito competitivo em que o padrão de serviço exigido pelos clientes é cada vez mais elevado, hoje tanto nos sectores privados ou públicos, os clientes esperam ser bem atendidas e com alguma rapidez. Por isso atualmente muitas empresas em cabo verde que possuem um grande número de clientela tem adotado os sistemas de gestão de filas de espera com o objetivo de tentar melhorar o seu atendimento e gerir melhor as suas filas tendo em conta o número elevado de pessoas que ficam dentro do estabelecimento esperando ser atendidas. Esses sistemas trazem muitas vantagens para as empresas, como exemplo, agora as filas estão muito mais bem organizadas, já não há aqueles constrangimentos de pessoas discutindo quem encontrou quem na fila, ou também pessoas que tentam furar as filas aproveitando a desorganização da mesma. Agora tudo é tratado com senhas que podem ser retiradas pelos clientes, assim cada um é atendido conforme o número que retirou.

São várias outras vantagens que foram trazidas por esses sistemas, mas também há que mencionar as limitações que tem sido claras por parte dessa tecnologia. Embora trouxeram mais organização para as filas, não tem conseguido diminuir significativamente o tempo de espera do cliente, o que tem trazido cada vez mais descontentamento por parte do público aqui em Cabo Verde.

Na minha ótica, uma excelente forma de tentar ultrapassar essa deficiência seria estender as funcionalidades desses sistemas para a computação móvel. Tendo em conta que agora quase todas as pessoas possuem telemóveis modernos que funcionam com sistemas operativos, seria uma ótima forma de tentar resolver esta questão de tempo de espera do cliente na fila com uma aplicação. E feitas algumas pesquisas acerca desse tema vi que ainda não foi desenvolvido nada do genero para dispositivos móveis aqui em Cabo Verde.

## **2.1. Filas de espera**

Uma fila ocorre sempre que a procura por um determinado serviço é maior que a capacidade do sistema de prover este serviço, ou seja a fila ocorre quando temos muitas pessoas que querem ser atendidas num determinado serviço mas temos poucos atendedores para esse serviço.

Segundo Claudia Perereira (2009) as filas de espera são um fenómeno corrente no dia-a-dia, onde existem clientes que desejam prestações de serviços (servidores que para serem utilizados é necessário que os clientes esperem e formem uma fila física ou conceptual) e que haja um número de clientes superior ao número de servidores, pois o servidor demora algum tempo a atender cada cliente (tempo de serviço), e este serviço termina quando o cliente se retira.

Os pontos de interesse da teoria das filas são as seguintes: o tempo de espera do cliente, o número de clientes na fila e a razão entre o tempo de espera e o tempo de prestação de serviço.

### **2.1.1. Sistemas de gestão de filas de espera**

Os sistemas de gestão de filas de espera são tecnologias que permitem as empresas ou organizações otimizarem os seus serviços prestados, trazendo benefícios tanto para quem é atendido, tanto para quem está atendendo. Sendo um pouco mais claro, essa tecnologia permite gerir as filas de espera no atendimento ao público de forma simples e prática, focando naquilo que realmente interessa que é de melhorar o processo de atendimento e a rapidez do serviço.

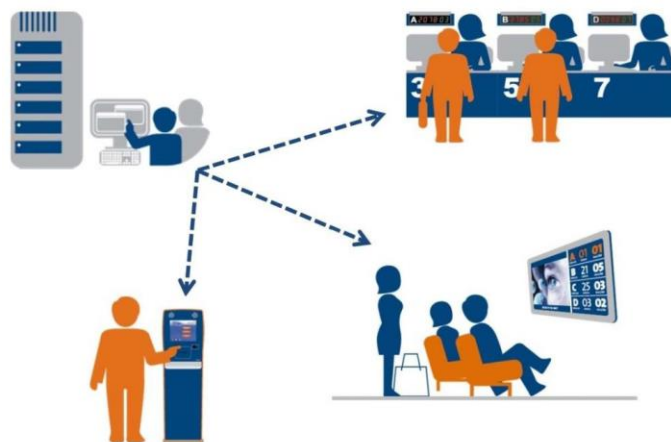
Nas empresas e instituições orientadas para prestação de serviços os clientes deparam-se frequentemente com o problema de terem longas filas de espera, resultando num decréscimo de eficiência e de produtividade. E a solução para isso esta nos sistemas de filas de espera que permitem o tratamento do tempo de espera, melhorando o nível dos serviços prestados e reduzindo o tempo de espera por parte do cliente, beneficiando ainda a sua imagem institucional perante os seus clientes.

### Vantagens de um sistema de gestão filas de espera:

- Aumento da qualidade do serviço prestado e do conforto do cliente em espera;
- Modernização do estabelecimento;
- Maior e melhor interação com os seus clientes;
- Otimização da gestão dos recursos utilizados para o atendimento;
- Aumentar a publicidade dos seus produtos e/ou serviços sem qualquer custo;
- Aumentar a satisfação do cliente.

#### 2.1.1.1. COMPONENTES DE UM SISTEMA DE FILAS DE ESPERA

Essa tecnologia é composta por um conjunto de softwares e hardware que enterragem entre si para o alcance dos objetivos propostos. Na figura a seguir podemos ter uma noção da composição desses sistemas.



*Figura 1-Componentes de um sistema de Filas de Espera*

Fonte -[http://sti-consultoria.com/site/pdf/fias\\_espera.pdf](http://sti-consultoria.com/site/pdf/fias_espera.pdf)

**Dispensador** – esse tem a função exclusiva de disponibilizar as senhas para os clientes num estabelecimento. Existem diferentes tipos de dispensadores, os que imprimem as suas senhas e os que utilizam rolo de senhas.

**Chamador** – Esse tem como principal objetivo chamar as senhas, e podem ser encontrados como uma aplicação instalada num computador (exemplo) ou como um dispositivo físico.

**Servidor** – Esse componente é responsável pelo controlo do sistema, nele pode se fazer toda a gestão necessária para que esse funcione bem para a seu estabelecimento.

**Display** – É aquela tela ou ecrã responsável por mostrar o andamento das filas. E também podem ser usados para conteúdos de publicitários da entidade.

#### 2.1.1.2. TIPOS DE SISTEMAS EXISTENTES

Feitos algumas pesquisas encontrei vários sistemas de gestão de filas de espera já criadas mas aqui só apresentarei alguns exemplos. Cada uma delas possui as suas determinadas características, mas o objetivo continua sendo o mesmo que é de gerir adequadamente as filas de espera. Temos então a seguinte lista:

### **QSystems**

Desenvolvido pela Altronix, segundo o seu site o sistema de Gestão de Filas de espera QSystems tem como finalidade melhorar o atendimento de uma forma simples e intuitiva, aliando qualidade e função a um preço acessível.

Com autonomia total e de fácil instalação, o sistema não necessita de computadores para funcionar, bastando ligar à corrente elétrica e utilizar os comandos de controlo. O QSystems reduz as filas de espera em qualquer área de atendimento, o que se traduz numa melhor experiência dos clientes a serem atendidos e em mais produtividade da equipa de balcão. Para além da ordenação do fluxo de pessoas, inclui a possibilidade de utilizar o display para mensagens informativas, tornando-se num excelente canal de comunicação com os clientes.

### **Vantagens**

- Melhora os tempos médios de espera sob 2 prismas distintos: o tempo de espera real e o tempo de espera percebido pelos clientes;



- Ordena sequencialmente, por hora de chegada, os clientes no espaço de atendimento;
- Melhor organização nas salas de espera, uma vez que os utentes sabem exatamente a altura em que serão atendidos;
- Reduz a frustração do cliente, o que se traduz numa diminuição das taxas de abandono nas filas de espera;
- Otimização do fluxo de entrada e saída de pessoas, tornando todos os serviços mais fluídos e sincronizados entre si;
- Fideliza o cliente, como resultado da satisfação criada pelo rápido atendimento;
- Otimização de processos internos, libertando tempo e recursos;
- Melhor gestão dos recursos humanos, o que se traduz em ganhos de produtividade;
- Maior capacidade de controlo e planeamento da direção;
- Modernização da sala de espera e atendimento;
- Permite comunicar informação importante aos utentes que chegam e aguardam a sua vez, através das mensagens no rodapé informativo;
- Promoção da marca diretamente no display e nas senhas de espera;
- Monitorização de estatísticas em tempo real sobre o serviço de atendimento;
- Reencaminhar facilmente os clientes para outros departamentos e serviços;
- Aumenta as vendas diárias e reduz custos;
- Instalação rápida e prática, com baixos custos de manutenção;
- Preço muito acessível, quando comparado com outras soluções no mercado;



Figura 2- Componentes da QSystems

Fonte: <http://www.qsystems.pt/>

## Qmachine

O Software Qmachine de acordo com o seu site, destaca-se no mercado pela sua funcionalidade e versatilidade, pois tem a capacidade de se adaptar às diversas necessidades e sectores. Desde a sua versão mais simplificada, onde se demonstra uma ferramenta útil para a ordenação de filas, até à versão mais complexa capaz de dar resposta a todas as exigências.

A sua funcionalidade permite a criação e configuração de layouts do sistema e parâmetros do sistema de uma forma muito intuitiva e acessível.

Caso exista um projeto específico e com funcionalidades que ainda não estão presentes na versão atual, a equipa de desenvolvimento QMACHINE está à disposição dos clientes para estudar a possibilidade de desenvolvimentos à medida.

A chamada dos clientes pode ser realizada através de botões físicos, no entanto a nossa empresa disponibiliza um software que é instalado nos PC's (postos de chamada/balcões) de forma a tornar o nosso sistema tecnologicamente dos mais avançados do mercado.

## Funcionalidades

- Melhorar qualidade do serviço;
- Melhorar imagem organizativa;
- Otimizar processos de atendimento;
- Reduzir aglomerados nas zonas de atendimento;
- Maior eficiência na gestão do atendimento;
- Divulgação de informação, formação, promoções aos clientes/utentes em espera;
- Planear atendimento baseado em informações dadas pelo sistema;
- Gestão de utilizadores/operadores (senha e password);
- Informação estatística detalhada;
- Informação para cliente/utente via WEB;
- Informação para cliente/ utente via SMS;
- Alertas para utentes em espera;
- Informação de tempo de espera;



*Figura 3-Q-magine*

Fonte: [http://www.q-magine.com/store\\_pt/index\\_2010\\_v2.php](http://www.q-magine.com/store_pt/index_2010_v2.php)

## **Sistema Inline da New Vision**

As vantagens da utilização de um sistema de Gestão de Filas de Espera são muitas, então segundo o site da INLINE, esse pode ajudar na:

### **Melhoria da imagem da organização:**

- Modernidade e eficiência;
- Satisfação dos utentes/clientes;
- Incremento do número de utentes/clientes que utilizam os serviços da organização;
- Venda da imagem/produtos da organização aproveitando o tempo de espera dos utentes/clientes.

### **Melhoria da qualidade efetiva do serviço prestado:**

- Demonstração de uma preocupação com o bem-estar dos utentes/clientes;
- Melhoria efetiva da qualidade do ambiente geral;
- Sentimento de tempos de espera mais curtos por parte do utente ou cliente;
- Respeito pelos utentes e pela confidencialidade nos assuntos tratados;

### **Melhoria no aproveitamento dos recursos humanos**

- Melhor ambiente de trabalho;
- Satisfação dos recursos humanos da organização;
- Distribuição equitativa do trabalho;
- Maior produtividade e eficácia.

### **Melhoria no aumento da Produtividade**

- Maior produtividade no atendimento devido ao ambiente de trabalho;
- Conhecimento rigoroso de diversos indicadores estatísticos;
- Gestão automática de prioridades no atendimento;
- Gestão em tempo real dos recursos humanos afetos ao atendimento;

## **O Inline funciona de uma maneira muito simples**

O cliente retira uma senha na dispensadora, espera a sua vez, um operador de balcão chamará o cliente pelo número da senha que simultaneamente irá aparecer no ecrã. Após o atendimento o cliente deixará a loja, deixando no sistema Inline toda a informação correspondente á sua senha.

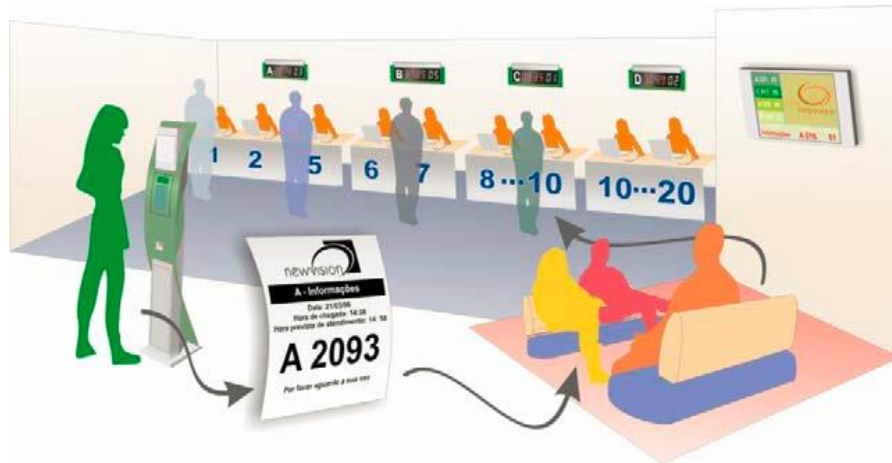


Figura 4-Funcionamento do sistema INLINE

Fonte -[http://sti-consultoria.com/site/pdf/fias\\_espera.pdf](http://sti-consultoria.com/site/pdf/fias_espera.pdf)

### Que informação guarda o Inline?

- Quando foi emitida a senha (dia, hora);
- Quanto tempo durou o atendimento;
- Foi atendido por quem;

### Como é que o Inline faz isso?

O Inline possui uma Base de Dados que emite os números das senhas e guarda toda a informação relacionada com cada número emitido. Todos os dados encontram-se guardados de modo a que sejam utilizados em formato de estatísticas. A comunicação entre os componentes utiliza a rede de infraestruturas instalada.

### Componentes

- Inline Database – Armazena todas as informações sobre a fila de espera e senhas;
- Inline Business – Componente central que integra todas as comunicações entre os diferentes componentes;

- Inline Hardware – Componente que comunica com os dispositivos de hardware e alguns componentes de software;
- Inline Manager – Interface de configuração e de gestão;
- Inline Web Client – Interface do operador para chamar senhas;
- Inline Displays – Software que permite a integração do Inline com um terceiro Signage Digital;
- Dispensadores – Dispositivo físico para distribuir senhas;

A figura a seguir mostra um exemplo de um display da Inline.



Figura 5- Exemplo de display da INLINE

Fonte: [http://sti-consultoria.com/site/pdf/fias\\_espera.pdf](http://sti-consultoria.com/site/pdf/fias_espera.pdf)

## Q.track

A solução Q.track segundo o seu site, é uma solução modular e de baixo custo que lhe permite gerir as filas de espera ou atendimento e conteúdos multimédia (TV corporate).

O Q.track, permite a gestão de vários locais com múltiplos serviços, fornecendo dados em tempo real através da web de forma, permite toda a gestão de conteúdos multimédia em tempo real e também fazer inquéritos interativos de satisfação ao cliente com resultados em tempo real.

## **Vantagens**

- Aviso automático de entrada de novas senhas;
- Atualização automática do número de senhas por atender;
- Relatórios em tempo real dos serviços de atendimento;
- Definição de regras de prioridade de atendimento;
- Os clientes são colocados numa fila virtual, permitindo-lhes a liberdade; para navegar ao redor da loja e olhar para outros produtos;
- Conhecimento de todos os indicadores de atendimento via web;
- Inserção de publicidade sobre produtos ou serviços aproveitando o tempo de espera dos clientes;
- Produção de relatórios sobre vários parâmetros (ex. tempo de utilização);
- Reduzir o tempo de espera do cliente e aumentar a produtividade dos funcionários;
- Aumento da satisfação dos clientes;

## **Benefícios**

- Melhoria da imagem da sua organização
- Redução do tempo de espera
- Mais qualidade no serviço prestado aos clientes
- Melhor aproveitamento dos recursos humanos
- Maior produtividade e eficácia dos recursos humanos
- Reencaminhamento de utentes entre serviços através de prioridades de atendimento por serviço

Na figura seguinte podemos ver o funcionamento da Q.Track.

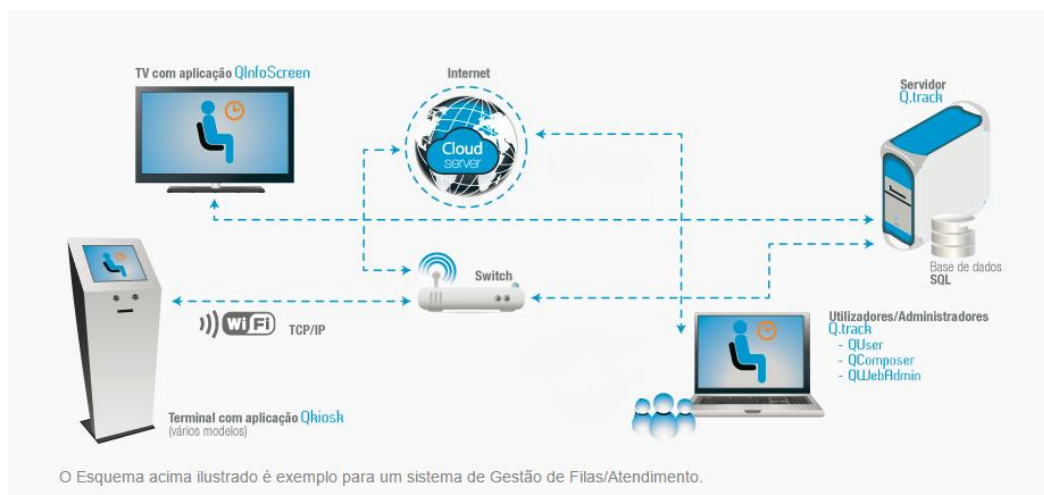


Figura 6- Funcionamento da Q.Track

Fonte: <http://q.track.pt>

Feita essa pesquisa deu pra ver que esses sistemas possuem todos os mesmos objetivos, o que diferencia uns dos outros são só as inovações ou novas funcionalidades implementadas em cada uma delas. Cada um é utilizado conforme a necessidade da empresa que adquiriu uma delas. Existem varias outras marcas dessa tecnologia mas trouxe apenas o necessário.

Em Cabo Verde muitas das empresas e instituições já utilizam alguns sistemas para controlo das suas filas de espera, algumas até que foram desenvolvidas aqui em Cabo-Verde por grupos de desenvolvedores segundo pesquisas feitas.

## 2.2. Android

Segundo Luciano Alves da Silva (Programando passo a passo 6ª Edição, 2013, Pag.6) o Android é uma plataforma desenvolvida pelo Google voltada para dispositivos móveis totalmente aberta e livre (Open Source), que foi divulgada em 5 de novembro de 2007. Inicialmente o sistema Android foi desenvolvido pelo Google mas atualmente essa plataforma é mantida pela OHA (Open Handset Alliance), um grupo constituído por aproximadamente 84 empresas, as quais se uniram para inovar e acelerar o desenvolvimento de aplicações e serviços, com o objetivo de trazer aos consumidores uma experiência mais rica em termos de recursos e menos dispendiosa no mercado móvel.



### 2.2.1. Estrutura Geral da plataforma Google Android

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.9) o Android SDK é uma ferramenta de desenvolvimento que disponibiliza um conjunto de API's necessárias para desenvolver aplicações para a plataforma Android, utilizando a linguagem Java.

#### Alguns dos recursos encontrados nessa plataforma:

- **Application framework:** Permite a reutilização e substituição de componentes;
- **Dalvik virtual machine:** É uma Máquina Virtual Java (JVM) voltada para dispositivos móveis;
- **Browser Integrado** baseado no webkit engine;
- **Gráficos Otimizados** O Android é constituído por bibliotecas 2D e 3D baseada na especificação OpenGL ES 1.0;
- **SQLite:** Sistema Gerenciador de Banco de Dados (SGBD) já embutido no Android para guardar dados;
- **Suporte multimídia:** A plataforma já oferece para áudio, vídeo e formatos de imagem (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF);
- **Telefonia GSM** (dependente de hardware);
- **Bluetooth, EDGE, 3G, e WiFi** (dependente de hardware) ;
- **Câmera, GPS, compasso, e acelerômetro** (dependente de hardware);
- **Rico ambiente de desenvolvimento**, incluindo um emulador de dispositivo, ferramentas de depuração, memória, performance e um plugin para o Eclipse (ADT);

#### 2.2.1.1. ARQUITETURA DO ANDROID

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.10) o android possui a seguinte estrutura como mostra a figura abaixo.



*Figura 7-Arquitetura plataforma Android*

*Fonte: <http://www.apostilaandroid.net/>*

#### 2.2.1.2. APLICAÇÕES

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.10) o Android nos fornece um conjunto de aplicações fundamentais, como por exemplo:

- Um cliente de e-mail;
- Programa de SMS;
- Agenda;
- Mapas;
- Navegador;
- Contatos, entre outros.

Todas essas aplicações acima mencionadas foram desenvolvidos na linguagem de programação Java.

#### 2.2.1.3. BIBLIOTECAS

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.11) o Android nos fornece um conjunto de bibliotecas C/C++ utilizadas por vários componentes do sistema. Veja algumas das bibliotecas abaixo:

- **System C library:** Consiste em uma implementação derivada da biblioteca C padrão baseado no sistema (libc) do BSD sintonizada para dispositivos rodando Linux;
- **Media Libraries:** Baseado no PacketVideo's OpenCORE; são as bibliotecas que suportam os mais diversos formatos de áudio e vídeo, incluindo também imagens;
- **Surface Manager:** Responsável pelo acesso ao subsistema de exibição bem como as múltiplas camadas de aplicações 2D e 3D;
- **LibWebCore:** Consiste em um web browser engine utilizado tanto no Android Browser quanto para exibições web;
- **SGL** – o engine de gráficos 2D;
- **3D libraries:** Uma implementação baseada no OpenGL ES 1.0 APIs; As bibliotecas utilizam aceleração 3D via hardware (quando disponível) ou o software de renderização 3D altamente otimizado incluído no Android;
- **FreeType** – Biblioteca responsável pela renderização de fontes bitmap e vetor;
- **SQLite** – Conforme já mencionado, consiste no sistema gerenciador de banco de dados (SGBD) relacional disponível para todas as aplicações;

#### 2.2.1.4. ANDROID RUNTIME

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.11) o Android é constituído por um conjunto de bibliotecas que fornece a maioria das funcionalidades disponíveis nas principais bibliotecas da linguagem Java.

Ainda no mesmo autor, toda aplicação Android executa em seu próprio processo, com sua própria instância da máquina virtual Dalvik. O Dalvik foi escrito de forma a executar várias VMs eficientemente. Ele executa arquivos .dex, que é otimizado para consumo mínimo de memória. A VM é baseada em registros e executa classes compiladas pela linguagem Java que foram transformadas em arquivos .dex, através da ferramenta “dx” incluída no SDK. O Dalvik VM foi baseado no kernel do Linux

para funcionalidades subjacentes como o encadeamento e a gestão de baixo nível de memória.

#### 2.2.1.5. LINUX KERNEL

Segundo Luciano Alves da Silva (Programando passo a passo, 2015, Pag.11) o Android foi projetado em cima da versão 2.6 do kernel do Linux para os serviços centrais do sistema, tais como segurança, gestão de memória, gestão de processos, entre outros. O kernel também atua como uma camada de abstração entre o hardware e o resto do software.

### 2.2.2. Componentes de uma aplicação

Segundo Carlos Silva & Pedro Veloso (introdução a programação, pag.14) As componentes de uma aplicação são os blocos básicos de construção da mesma. A SDK do Android fornece vários blocos diferentes, sendo que através de cada um destes o sistema consegue interagir com sua aplicação. Nem todos esses têm um carácter de interação direta do utilizador como poderá ver em seguida. Uma aplicação pode ser constituída por vários destes blocos, e estes blocos são comunicáveis entre si na maioria dos casos. Em seguida podemos ver alguns dos componentes de uma aplicação.

- **Activity**

Uma Activity é um ecrã da aplicação (ou form noutras linguagens ou sistemas). É aqui que habitualmente se produz e elabora a componente de interação visual de uma aplicação.

- **Service**

Um Service como o nome já diz é um serviço que pode ser criado. Ele não tem um frontend visual como a Activity, e são utilizados normalmente para executar tarefas mais demoradas em plano de fundo. Geralmente é usado como um suporte às várias Activities de uma aplicação, tratando de assuntos como a comunicação com a ligação à Internet, atualizações periódicas de conteúdos, e intercomunicação entre aplicações.

- **Broadcast Receiver**

Um Broadcast Receiver é uma componente que reage a eventos ou sinais de sistema. No processo de instalação de uma aplicação em Android, o instalador procura pelas componentes deste tipo e insere numa tabela de sistema as referências sobre a que tipo de eventos aquela aplicação reage. Entre eventos deste tipo enviados pelo sistema operativo podem-se contar coisas como a mudança da hora atual, alteração do estado da bateria, alteração das condições de conectividade entre muitas outras coisas. Desse modo é possível produzir uma aplicação que tenha uma componente desse tipo e que cada vez que o sistema envia um sinal de que se perdeu ligação à rede, é adiada a atualização de conteúdos via rede até que o sinal de ligação restabelecida seja enviado e mais uma vez recebido pela nossa componente de Broadcast Receiver, desse modo fazendo uma gestão inteligente de quando é que a aplicação vai executar um código relativo à tentativa de obtenção de atualizações periódicas.

- **Content Provider**

Um Content Provider é uma forma de providenciar informação da nossa aplicação a outras aplicações instaladas no sistema operativo. Isto permite criar um ecossistema muito poderoso assim que o leitor de aperceber de que forma esta componente pode ser utilizada e das diversas aplicações que atualmente já tiram ou permitem tirar partido desta capacidade. Um exemplo de uma aplicação com uma componente deste género é a aplicação de Contactos presente na generalidade de terminais Android. Esta aplicação fornece um Content Provider para que qualquer outra aplicação possa saber quais os contactos guardados no dispositivo, e iterar sobre a listagem dos mesmos.

A componente de Content Provider fornece deste modo uma forma geral e abstrata de fornecer os conteúdos da aplicação em que se insere, esta forma genérica segue sempre a mesma estrutura tornando fácil o uso desta componente por outras aplicações. Ao mesmo tempo, a implementação de como a informação que está a ser fornecida é da escolha do programador, que pode mesmo controlar que tipo de informação é fornecida e a forma como esta está guardada no dispositivo (podendo mesmo estar na nuvem ao invés de fisicamente no dispositivo por exemplo).

### 2.2.3. Ciclo de vida de uma aplicação

Segundo Carlos Silva & Pedro Veloso (introdução a programação, pag.39) o Android é um sistema operativo multitarefa, e tendo sido concebido nesse sentido desde o início do seu desenvolvimento consta de uma gestão inteligente e eficaz de processos, sendo que isto causa consequências à forma como se programam aplicações na plataforma.

As Activities tem um conjunto de métodos de callback que são utilizados na construção lógica das aplicações. O diagrama apresentado de seguida pela figura 8 mostra o ciclo de vida de uma Activity e os métodos que callback (onStart, onPause, etc) que são chamados aquando á ocorrência de cada uma dessas condições.

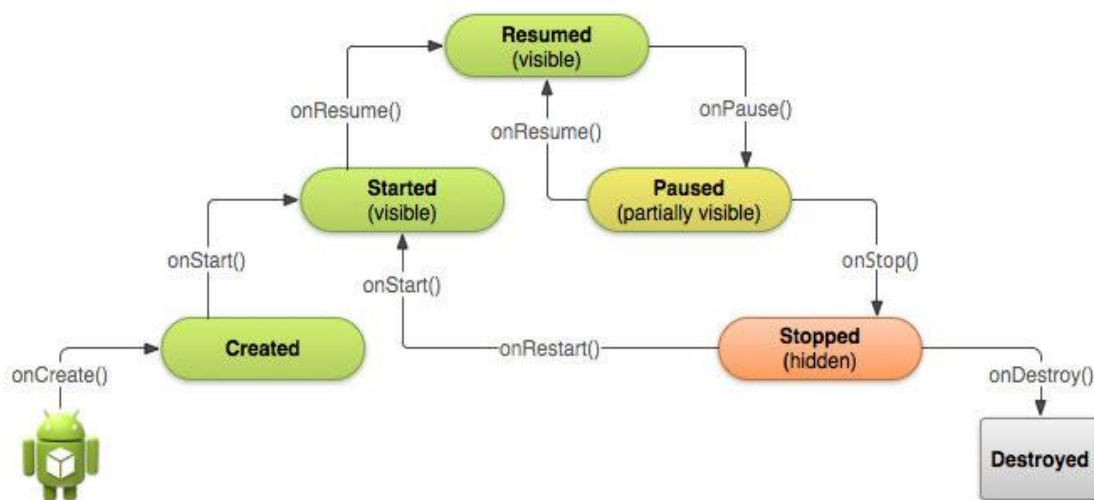


Figura 8-Ciclo de vida de uma Activity

Fonte: Livro Google android, Crie aplicações para celulares e tablets.

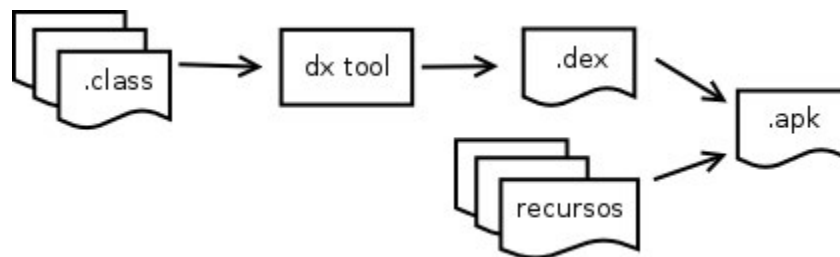
### 2.2.4. A execução das aplicações

Segundo João Monteiro (Google android, Crie aplicações para celulares e tablets) as aplicações implementadas utilizando a linguagem Java são executadas através de uma máquina virtual, baseada em registradores e otimizada para consumir pouca memória, denominada de Dalvik. Ao contrário da máquina virtual Java que executa bytecodes, a Dalvik utiliza arquivos no formato .dex gerados a partir de classes Java compiladas. Esta conversão é feita pela ferramenta dx que acompanha o Android SDK.

Basicamente, o que é feito é o agrupamento de informações duplicadas que se encontram espalhadas em diversos arquivos `.class` em um arquivo `.dex`, com o tamanho menor do que os arquivos que a originaram. O `dx` também faz a conversão de bytecodes para um conjunto de instruções específicas da máquina virtual Dalvik.

Depois de criado, o arquivo `.dex` e todos os recursos utilizados na aplicação, como imagens e ícones, são adicionados em um arquivo `.apk`, que é a aplicação em si, capaz de ser instalado em um dispositivo. Estes arquivos se encontram na pasta `bin` do projeto.

É possível distribuir sua aplicação para outras pessoas apenas fornecendo o arquivo `.apk`. A figura 9 a seguir demonstra o processo de geração do aplicativo.



*Figura 9- Processo de geração de aplicações*

*Fonte: Livro Google android, Crie aplicações para celulares e tablets.*

Ainda seguindo o mesmo autor, no sistema operacional Android, para cada aplicação é atribuído um usuário único de sistema e apenas este usuário recebe permissões para acessar seus arquivos. Além disso, por padrão, cada aplicação é executada em um processo próprio, que possui também sua própria instância da máquina virtual Dalvik. Sendo assim, ela é executada de forma segura e isolada das outras aplicações.

Neste contexto, uma aplicação não pode acessar arquivos de outra e tão pouco a acessar diretamente recursos do sistema operacional como a lista de contatos, câmera, gps e rede, sem que o usuário explicitamente autorize o acesso durante a instalação dela.

### 2.2.5. Versões recentes

O android tem evoluído muito ao longo dos anos apresentando melhorias desde a primeira versão, Android 1.0. Segundo Carlos Silva e Pedro Veloso (Introdução a programação android) esta versão do Android foi distribuída na mesma altura que houve o início da venda do HTC G1, o primeiro smartphone comercial com Android. A sua data de disponibilidade para o público é portanto coincidente com a comercialização efetiva do dispositivo que teve lugar em Outubro de 2008. Várias versões foram lançadas desde então.

#### 2.2.5.1. ANDROID 6.0 MARSHMALLOW

Dando um salto para o Android 6.0 (Marshmallow) segundo pesquisas, esta versão tem como destaque principal as melhorias na usabilidade do sistema. Uma prova disso é que a interface ficou ainda melhor esteticamente, com animações mais sutil ou suave e design refinado que melhoram a fluidez das tarefas.

Com o recurso *Now On Top* você pode ver informações úteis e aplicativos rapidamente apenas apertando um botão. E a bateria ganhou o recurso *Doze* que melhora ainda mais o seu desempenho.

A versão Marshmallow também recebeu melhorias em relação à segurança, oferecendo permissões de privacidade para os aplicações e a possibilidade de utilizar o leitor de digitais para desbloquear e autorizar algumas tarefas.

#### 2.2.5.2. ANDROID NOUGAT (7.0 E 7.1)

Segundo pesquisas o Android 7.0 Nougat foi anunciado pelo Google em setembro de 2016 e é a atual versão principal do sistema operacional Android. Primeiramente lançado como uma versão beta em 9 de março de 2016, foi lançado oficialmente em 22 de agosto de 2016, com os dispositivos Nexus sendo os primeiros a receberem a atualização.

O Nougat introduz mudanças notáveis para o sistema operacional e sua plataforma de desenvolvimento, incluindo a capacidade de exibir vários aplicativos na tela de uma só vez em uma visualização em tela dividida, suporte a respostas direto das



notificações, bem como um ambiente e suporte Java baseado em OpenJDK para a API de renderização de gráficos Vulkan, e atualizações do sistema "seamless" em dispositivos suportados.

Com a chegada do Pixel e do Pixel XL, a Google começou a trabalhar na versão 7.1.1 para os Nexus, que foi liberada no dia 5 de dezembro. Agora, o time de desenvolvedores do Android liberou a versão oficial do Android 7.1.2 Nougat.

Depois de passar algum tempo em teste desde o final de janeiro deste ano a Google começou a liberar o update para a versão Android 7.1.2 para a linha Pixel, bem como para todos os aparelhos do Google anunciados depois de 2015. De acordo com uma publicação dos administradores da Comunidade de utilizadores do Píxel, Orrin, essas são as mudanças mais significativas adicionadas ao sistema operacional:

- Otimização da leitura das impressões digitais;
- Otimização da conexão Bluetooth;
- Adição de alertas sobre a bateria.

Além dos destaques acima, também tiveram correções relacionadas a problemas com o desligamento automático dos aparelhos, à listra rosa que aparecia na câmara de alguns utilizadores, e melhoria da saída de som.

### 3. Tecnologias Utilizadas

#### 3.1. Linguagem JAVA

As linguagens de programação podem ser consideradas instrumentos fundamentais para o desenvolvimento de aplicações. Desta forma, o profissional não deve restringir seu conhecimento a somente uma linguagem, e sim buscar descobrir as vantagens e desvantagens oferecidas por cada uma, tendo em vista que não há melhor ou pior, e sim a que mais irá se adequar ao seu projeto (MATTOS, 2004).

Segundo a Deitel (Java Como Programar, 2012) Java é uma linguagem de programação orientada a objeto, de alto nível, com sintaxe similar a do C++, que foi lançada em 1991, pela Sun Microsystem. Surgiu a partir de um projeto, chamado “Projeto Green”, que foi desenvolvido por um grupo de pesquisadores, onde o objetivo não era simplesmente criar uma linguagem de 4 programação convencional, mas sim determinar uma nova tendência do mundo digital.

A princípio foi destinada à programação de dispositivos eletrônicos inteligentes voltados ao consumo popular, pois a Sun acreditava na computação embarcada. Seu advento veio somente com o surgimento da internet (SILVEIRA, 2003).

A linguagem Java é considerada hoje uma referência no mercado de desenvolvimento de softwares e tem seu ambiente de execução presente em diversas aplicações como: web, desktop, servidores, mainframes, jogos, aplicações móveis, chips de identificação entre tantos outros. (JAVA, 2008).

De acordo com Pamplona (2008), Java pode ser dividida em três grandes edições:

- a) Java Standard Edition (J2SE), que é a tecnologia voltada para computadores com poder de processamento e memória considerável;
- b) Java 2 Micro Edition (J2ME), tecnologia voltada para dispositivos móveis, com limitação de processamento ou memória;
- c) Java 2 Enterprise Edition (J2EE), tecnologia para aplicações corporativas, que podem estar na internet ou não.

### **Principais características:**

- Suporte à orientação a objetos;
- Portabilidade;
- Segurança;
- Linguagem Simples;
- Alta Performance;
- Dinamismo;
- Interpretada (o compilador pode executar os bytecodes do Java diretamente em qualquer máquina);
- Distribuído; Independente de plataforma;
- Tipada (deteta os tipos de variáveis quando declaradas);

Nesse projeto o JAVA foi utilizado para o desenvolvimento da aplicação móvel pelo fato de ser uma referência no mercado de desenvolvimento de softwares e também por ser a linguagem que foi adotada pela Google para o desenvolvimento em Android até a data.

### **3.2. Android Estúdio**

Segundo o Google (2013) o Android Studio é um novo e totalmente integrado ambiente de desenvolvimento, que foi recentemente lançado pela Google para o sistema operativo Android. Foi desenhado para fornecer novas ferramentas para o desenvolvimento de apps e para fornecer uma alternativa ao Eclipse, atualmente o mais amplamente utilizado IDE. Além do editor de código e das ferramentas de desenvolvedor avançados do IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na criação de aplicativos Android, como:

- Um sistema de compilação flexível baseado no Gradle;
- Um emulador rápido com inúmeros recursos;
- Um ambiente unificado para você poder desenvolver para todos os dispositivos Android;
- Instant Run para aplicar alterações a aplicativos em execução sem precisar compilar um novo APK;

- Modelos de códigos e integração com GitHub para ajudar a criar recursos comuns dos aplicativos e importar exemplos de código;
- Ferramentas e estruturas de teste cheias de possibilidades;
- Ferramentas de verificação de código suspeito para detectar problemas de desempenho, usabilidade, compatibilidade com versões e outros;
- Compatibilidade com C++ e NDK;
- Compatibilidade embutida com o Google Cloud Platform, facilitando a integração do Google Cloud Messaging e do App Engine;

### 3.2.1. ANDROID SDK

Segundo Ricardo R. Lecheta (Android Essencial, Versão resumida do livro da Google Android, 2016, Pag. 33) o android SDK é o software utilizado para desenvolver aplicações no android, que tem um emulador para simular o dispositivo, ferramentas utilitárias e uma API completa para linguagem Java, com todas as classes necessárias para desenvolver as aplicações.

Diferente do Eclipse o Android Studio já vem com o SDK incluído nele por isso não é necessário baixá-lo.

#### **Requisitos de sistema**

Segundo Neil Smyth (Android Studio Development Essentials – Android 6 Edition, 2015) o desenvolvimento de aplicações para Android utilizando o Android estúdio e o SDK pode ser executado em qualquer um dos seguintes tipos de sistema:

- Windows Vista (32 bits ou 64 bits);
- Windows 7 (32 bits ou 64 bits);
- Windows 8 / Windows 8.1 ou posterior;
- Mac OS X 10.8.5 ou posterior (somente sistemas baseados em Intel);
- Sistemas Linux com versão 2.15 ou posterior da Biblioteca GNU C (glibc);
- Mínimo de 2GB de RAM (4GB é preferido);
- Aproximadamente 4,5 GB de espaço disponível em disco;

### **3.3. Linguagem PHP**

Segundo MAURICIO BARRETO (2000) PHP (Hypertext Preprocessor) é uma linguagem que permite criar sites WEB dinâmicos, possibilitando uma interação com os utilizadores através de formulários, parâmetros do URL e links. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas html puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

O que diferencia PHP de um script CGI escrito em C ou Perl é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o script CGI (Common Gateway Interface) gere todo o código HTML, ou leia de um outro arquivo.

Segundo a O'RELLY (PHP CookBook, 2014) O PHP é o motor por trás de muitas de aplicações web dinâmicas. Seu conjunto de características, sintaxe acessível e suporte para diferentes sistemas operacionais e servidores tornaram-na uma linguagem ideal tanto para o desenvolvimento rápido da web como para a construção metódica de sistemas complexos.

Uma das razões principais para o sucesso do PHP como uma linguagem de script web é a sua origem como uma ferramenta para processar formulários HTML e criar páginas web. Isso torna o PHP muito amigável na web. Além disso, ele é ansiosamente promíscuo quando se trata de aplicações externas e bibliotecas. O PHP pode falar com uma infinidade de bancos de dados, e conhece inúmeros Protocolos. PHP também torna simples analisar dados de formulário e fazer solicitações HTTP.

Para esse projeto utilizei o PHP para criar a Web Service que irá permitir a comunicação entre a aplicação e o servidor que contem a base de dados do sistema de gestão de fila de espera. E na Website foi utilizado na parte de processamento de login, cadastro e para encaminhar os formulários preenchidos para a sua base de dados. Também foi utilizado na parte do administrador no gerenciamento de

utilizadores, pedidos e para confirmar os pedidos através de envio de email que é encaminhado para o destino através do PHP.

### **3.4. Base de Dados MySQL**

Segundo Robin Nixon (O'RELLY, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition, Pag 171) um banco de dados é uma coleção estruturada de registros ou dados armazenados em um sistema organizados de forma a poderem ser rapidamente pesquisados e as informações recuperadas.

O SQL no MySQL significa Structured Query Language. Esta linguagem é vagamente baseado em Inglês e também usado em outros bancos de dados como Oracle e Microsoft SQL Server. Ele é projetado para permitir solicitações simples de um banco de dados através de comandos.

Segundo André Milani (MySQL - Guia do Programador, 2007) o MySQL é um banco de dados completo, robusto e extremamente rápido, com todas as características existentes nos principais bancos de dados disponíveis no mercado.

Uma de suas peculiaridades são suas licenças para uso gratuito, tanto para fins estudantis como para realização de negócios, possibilitando que empresas o utilizem livremente.

#### **Características**

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Alta compatibilidade com linguagens como PHP, Java, Python, C#, Ruby e C/C++;
- Baixa exigência de processamento (em comparação como outros SGBD);
- Vários sistemas de armazenamento de dados (batabase engine), como MyISAM, MySQL Cluster, CSV, Merge, InnoDB, entre outros;
- Recursos como transactions (transações), conectividade segura, indexação de campos de texto, replicação, etc.
- Instruções em SQL, como indica o nome.

A escolha de usar o MYSQL no meu projeto foi por causa das características descritas a cima, e também pelo fato de que em algumas aulas durante o curso, tivemos algum contato com ele, e já se conseguia fazer alguma coisa utilizando-o ou no WampServer ou no XampServer, por tanto ficaria muito mais fácil apenas aprofundar um pouco mais acerca dessa tecnologia. É um software livre que interage muito bem com a linguagem PHP. Foi utilizado como banco de dados da Website e também é onde vou buscar os dados das filas de esperas com o Webservice criado com PHP.

### **3.5. Base de dados SQLite**

Segundo a K19 (Desenvolvimento Mobile Com Android, 2012, pág. 41) O SQLite é um banco de dados bem simples que consome poucos recursos, bastante usado em alguns dispositivos atualmente. Para utilizar o SQLite, é necessário que se crie uma subclasse de SQLiteOpenHelper. Em seguida é necessário criar os métodos onCreate() e onUpgrade(). O primeiro é chamado quando ainda não existe um banco de dados, nele você deve incluir os comandos para criar tabelas e inicializar qualquer tipo de dados, se preciso. O segundo é chamado quando a versão da base de dados é alterada, e nele você deve incluir quaisquer comandos relacionados à alteração do esquema, como alterações em tabelas e colunas.

O SQLiteOpenHelper oferece dois métodos que serão muito usados, o getWritableDatabase() e getReadableDatabase(). Como o nome indica, estes métodos servem para obter uma instância da base de dados. Estes métodos retornam uma instância de SQLiteDatabase, que é utilizada para fazer consultas aos dados.

Os métodos que são usados com esse propósito são o insert(), update() e delete(). Também são usados os métodos query() e rawQuery(). O primeiro oferece uma interface para criar consultas, enquanto o segundo permite utilizar SQL diretamente.

Segundo Damon oehlman & Sebastien Blanc (Pro Android Web Apps: Develop for Android Using HTML5, CSS3 & JavaScript, 2011, Pag. 6) O SQLite está rapidamente se tornando o padrão de fato para os bancos de dados embutidos ou seja bancos de dados internos. Ter acesso a um banco de dados leve, tal como o SQLite no cliente, torna-se possível criar aplicações que podem tanto armazenar como

utilizar cópias de informação de localização em cache, que normalmente poderiam ser armazenadas em um servidor remoto.

Tendo em conta tudo isso inseri o SQLite no meu projeto na parte de cadastro e login na aplicação, não vi necessidade de ter um banco de dados externo para esse propósito, serviu perfeitamente.

### **3.6. WebService RESTful**

Segundo a Deitel (Java Como Programar, 2012, pag.1020) o Representational State Transfer (REST) é um estilo arquitetônico de implementar Web Services. Esses costumam ser chamados de Web Services RESTful. Embora o próprio REST seja implementado utilizando o padrão Web. Cada método em um serviço Web RESTful é identificado por um URL único. Assim, quando o servidor receber uma solicitação ou requisição de um algum serviço, ele saberá imediatamente que operação realizar. Esses Web Services podem ser utilizados em uma aplicação ou ate mesmo diretamente de um navegador web. Os resultados de uma determinada operação podem ser armazenados em cache localmente pelo navegador quando o serviço é invocado com a solicitação GET. Isso pode tornar solicitações subsequentes a mesma operação mais rápidas carregando o resultado diretamente na cache do navegador.

WebServices RESTful são alternativas aqueles implementados com SOAP. Diferentemente dos serviços baseados em SOAP, a solicitação e resposta do REST não são empacotados em envelopes. O REST também não esta limitado a retornar dados somente no formato XML. Ele pode utilizar vários formatos, como XML, JSON, HTML, texto sem formatação e arquivos de Mídias.

Segundo João Bosco Monteiro (Google Android, Crie aplicações para celulares e tablets, pag 172) o REST consolidou-se como a WebServices padrão quando se trata de disponibilizar serviços na web e possui possuem basicamente cinco premissas:

- Utilizar os métodos do protocolo HTTP para representar as operações que pode ser executadas pelo serviço;
- Expor as informações através de URLs representativas, similar a uma estrutura de diretórios;



- O serviço não deve armazenar estado entre requisições;
- Transmitir os dados em formato XML e/ou JSON;
- O uso de hipermídia para representar possíveis transições;

Tendo em conta esses aspetos mostrados, optou-se pela utilização de um Webservice com arquitetura RESTful na aplicação para estabelecer a comunicação entre a aplicação e uma base de dados remota, visto que é muito mais simples de implementar porque possui menos burocracia comparado com o SOAP. Segundo pesquisas, hoje em dia muitas aplicações estão utilizando essa arquitetura moderna.

### 3.7. Json

Segundo a Deitel (Java Como Programar, 2012, pág. 1020) a JavaScript Object Notation (JSON) é uma alternativa a XML para representar dados. JSON é um formato de troca de dados baseados em textos utilizando para representar objetos em JavaScript como coleções de pares nome/valor representados como Strings. Ele é comumente utilizado em aplicações AJAX. JSON é um formato simples que facilita a leitura, criação e análise de objetos e, como é muito menos extenso que a XML, permite que os programas transmitam dados eficientemente pela internet. Cada objeto JSON é representado como uma lista de nomes e valores de propriedade entre colchetes, no seguinte formato:

```
{
  "aluno": [
    {"nome": "João", "provas": [ { "nota": 8 }, { "nota": 6 }, { "nota": 10 } ] },
    {"nome": "Maria", "provas": [ { "nota": 3 }, { "nota": 5 }, { "nota": 8 } ] },
    {"nome": "Pedro", "provas": [ { "nota": 7 }, { "nota": 6 }, { "nota": 6 } ] }
  ]
}
```

*Figura 10-Exemplo de estrutura Json*

*Fonte: livro JAVA como Programar 8 edição*

A mesma estrutura ficaria assim no XML:

```

1  <?xml version='1.0' encoding='utf-8'?>
2  <aluno>
3      <nome> João </nome>
4      <provas>
5          <prova nota=8/>
6          <prova nota=6/>
7          <prova nota=10/>
8          <prova nota=2/>
9      </provas>
10 </aluno>
11 <aluno>
12     <nome> Maria </nome>
13     <provas>
14         <prova nota=3/>
15         <prova nota=5/>
16         <prova nota=8/>
17     </provas>
18 </aluno>
19 <aluno>
20     <nome> Pedro </nome>
21     <provas>
22         <prova nota=7/>
23         <prova nota=6/>
24         <prova nota=6/>
25     </provas>
26 </aluno>
27 </aluno>
28 </aluno>
29

```

Figura 11-Exemplo de Estrutura XML

Fonte: livro JAVA como Programar 8 edição

Muitas linguagens de programação agora suportam o formato de dados JSON. Uma extensa lista de bibliotecas JSON classificadas por linguagem podem ser localizadas em [www.json.org](http://www.json.org).

Nesse meu projeto utilizei o JSON como formato de dados, e está incluído na parte da recepção de dados, entre a aplicação e o Web Service.

### 3.8. Genymotion

O Genymotion que foi desenvolvido pela Genymobile é um emulador Android que inclui um conjunto completo de sensores e para interagir com um ambiente virtual Android. Com o ele você pode testar Aplicações do Android em uma ampla gama de dispositivos virtuais para desenvolvimento e demonstração.

Ele é rápido, simples de instalar e poderoso graças aos widgets de sensor de fácil utilização e Interação.

#### Requisitos para instalação:

Ele é compatível com os seguintes sistemas operativos:

- Microsoft Windows 7, 8 / 8.1, 10 (32 ou 64 bits);
- Mac OS X 10.9 ou superior;
- Linux Ubuntu 16.04 (Xenial Xerus) (64 bits) ou superior;
- Linux Debian 8 (Jessie) (64 bits);
- Linux Fedora 25 (64 bits).

## **Hardware**

O Genymotion requer que você tenha o seguinte hardware em seu computador:

- OpenGL 2.0 placa de vídeo capaz, com um driver atualizado. Recomendamos o uso de um dos seguintes chips gráficos mínimos: Intel HD Graphics 4000 (2012), Nvidia GeForce 500 Série (2011), Série ATI Radeon HD 6000 (2011);
- CPU de 64 bits, com capacidade VT-x ou AMD-V, ativada nas configurações da BIOS;
- Memória RAM: pelo menos 2GB;
- Resolução de tela superior a 1024 x 768 pixels;
- Espaço livre no disco rígido: pelo menos 100MB.

## **Software**

Para executar dispositivos virtuais que poderá baixar partir do seu Genymotion, você deve instalar o Oracle Virtual Box. O site oficial do Genymotion recomenda por questões de performance que seja usada a versão 4.3.12.

## **Web browser**

Para instalação, atualizações e serviços fornecidos pelo site Genymotion, é necessário:

- Internet Explorer: versão 10 ou superior;
- Mozilla Firefox: versão 3.0 ou superior;
- Google Chrome: versão 2.0 ou superior;
- Safari: versão 4.0 ou superior.

Para o meu projeto em si o Genimotion foi utilizado para instalar e testar a minha aplicação, assim não foi preciso um telemóvel com android para fazer a depuração e realizar essa tarefa.

### **3.9. Dream Weaver**

O Dreamweaver é um programa de edição de páginas da internet, onde se pode trabalhar com imagens, textos, e muitos outros elementos para a Web.

O programa permite a criação e edição de páginas na internet, sendo em alguns aspetos, muito parecido com programas de criação de layouts. Com essa ferramenta não é necessário criar um código de programação, acelerando assim o tempo de produção de sites, embora seja muito complicado trabalhar com este programa sem conhecimentos básicos de layout de sites. O Dreamweaver tem suporte á maioria das linguagens estruturadas do mercado. O programa cria códigos na linguagem desejada, e o utilizador só precisa utilizar a interface gráfica. O usuário pode colocar uma imagem no seu site, e o programa escreve o código para a página da Web.

Um dos problemas com o Dreamweaver é que algumas funções criam os chamados “códigos sujos”. Estes são códigos com linhas em excesso, que a princípio não atrapalham o site. Porém, se um programador for olhar o código, para editar algo, ou corrigir algum problema, estes códigos sujos atrapalham, dificultando a edição.

Este programa é muito útil para os criadores de sites, porém é sempre bom que a pessoa aprenda as linguagens de programação, para fazerem seus próprios códigos, assim o Dreamweaver se torna um programa para consultar, e lembrar de alguns códigos, durante a programação. Algumas linguagens mais usadas no Dreamweaver, na versão CS6 são:

- ActionScript;
- Active Server Pages (ASP);
- ASP.NET;
- C#;
- Cascading Style Sheets (CSS);
- ColdFusion;
- EDML;

- Extensible HyperText Markup Language (XHTML);
- Extensible Markup Language (XML);
- Extensible Stylesheet Language Transformations (XSLT);
- HyperText Markup Language (HTML);
- Java;
- JavaScript;
- JavaServer Pages (JSP);
- PHP: Hypertext Preprocessor (PHP);

## **Requisitos de sistema da Dreamweaver CS6**

### **Windows:**

- Processador Intel® Pentium® 4 ou AMD Athlon® 64;
- Microsoft® Windows® XP com Service Pack 3 ou Windows 7 com Service Pack 1. Adobe® Creative Suite® 5.5;
- 512 MB de RAM;
- 1 GB de espaço livre em disco para a instalação; requer espaço livre adicional durante a instalação (não é possível instalar em dispositivos de armazenamento removíveis flash);
- Monitor de 1280 x 800 com placa de vídeo de 16 bits;
- Java™ Runtime Environment 1.6 (incluso);
- Unidade de DVD-ROM;
- Software QuickTime 7.6.6 necessário para reprodução de mídia HTML5;
- Este software não funciona sem ativação. Além de conexão de banda larga com a Internet, é necessário fazer o registro para ativação do software, validação de assinaturas e acesso a serviços online;

### **Mac OS**

- Processador Intel multicore;
- Mac OS X v10.6.8 ou v10.7. Os aplicativos da Adobe Creative Suíte 5, CS5.5 e CS6 são compatíveis com o Mac OS X v10.8 e v10.9 quando instalados em sistemas baseados em Intel;

- 512 MB de RAM;
- 1,8 GB de espaço livre em disco para a instalação; requer espaço livre adicional durante a instalação (não é possível instalar em um volume que use um sistema de arquivos que diferencia maiúsculas e minúsculas ou em dispositivos de armazenamento removíveis flash);
- Monitor de 1280 x 800 com placa de vídeo de 16 bits;
- Java Runtime Environment 1.6;
- Unidade de DVD-ROM;
- Software QuickTime 7.6.6 necessário para reprodução de mídia HTML5;
- Este software não funciona sem ativação. Além de conexão de banda larga com a Internet, é necessário fazer o registro para ativação do software, validação de assinaturas e acesso a serviços online;

Para o meu projeto utilizei essa ferramenta para desenvolver todo o ambiente da Website incluindo a parte do administrador. É um programa bastante útil por que ajuda a poupar algum tempo que seria desperdiçado caso tivesse que ir programar tudo.

### **3.10. Linguagem HTML**

Quando entramos numa página web qualquer, normalmente estamos interessados em ver toda a informação ou o conteúdo dessa página. Essa informação pode estar na forma de texto, imagem ou vídeo por exemplo. Então esse conteúdo da página web é definido com a linguagem HTML (HyperText Markup Language).

Segundo a K19 (Desenvolvimentos Web com HTML, CSS E Javascript, 2014, Pag. 21) HTML é uma linguagem de marcação originalmente proposta por Tim Berners-Lee no final da década de 1980. O seu era criar um mecanismo simples que pudesse ser utilizado por qualquer pessoa que quisesse disseminar documentos científicos.

Desde sua proposta até os dias de hoje, a linguagem HTML sofreu diversas alterações. A cada versão, novos recursos são adicionados e problemas corrigidos. A versão mais atual da especificação da linguagem HTML é a 5.

Segundo a Caelum o HTML é um conjunto de tags responsáveis pela marcação do conteúdo de uma página no navegador.

### **Versão HTML5**

Segundo Francisco Morais (HTML5 e CSS3 Domine a Web do futuro, 2012, Pag.27) o surgimento do HTML5 mudou muitas coisas no mundo do desenvolvimento web, com novos elementos, novas funcionalidades e diversas outras novidades que possibilitam experiências melhores e integrações que antes eram apenas desejos e sonhos dos desenvolvedores por aí. Mesmo com algumas funcionalidades ainda em processo de definição, já podemos aproveitar diversas das novidades que o HTML5 trouxe para o mundo. Algumas que são referente ao código que escrevemos: novos elementos, mudanças de sintaxe e atributos personalizados.

Toda a parte de visual da Website foi desenvolvido utilizando o HTML.

### **3.11. Linguagem CSS**

Segundo a K19 (Desenvolvimentos Web com HTML, CSS E Javascript, 2014, Pag.141) quando não definimos uma formatação específica para os elementos HTML de uma página web, eles são exibidos com o estilo determinado pelo navegador utilizado. Há dois problemas principais em deixar os navegadores decidirem qual formatação deve ser adotada. O primeiro é que cada navegador pode adotar uma formatação diferente. Consequentemente, uma mesma página web não será exibida exatamente da mesma forma em todos os navegadores. O segundo problema é que as formatações adotadas pelos principais navegadores não são “bonitas”.

Podemos padronizar a forma que as páginas web são exibidas nos diferentes navegadores e obter um visual agradável definindo a nossa própria formatação. O estilo das páginas deve ser definido com a linguagem CSS (Cascading Style Sheets).

### **Versão CSS3**

CSS3 é o mais recente padrão para CSS e segundo a W3schools CSS3 foi dividido em módulos. Ele contém a especificação CSS antiga (que foi dividida em pedaços menores). Além disso, novos módulos são adicionados. Alguns dos mais importantes módulos CSS3 são:

- Seletores;
- Modelo de Caixa;
- Fundos e Fronteiras;
- Valores de imagem e conteúdo substituído;
- Efeitos de texto;
- Transformações 2D / 3D;
- Animações;
- Layout múltiplo de colunas;
- Interface de usuário;

Segundo Francisco Morais (HTML5 e CSS3 Domine a Web do futuro, 2012) diversas adições do CSS são extremamente úteis para substituir vários tipos de imagens que utilizávamos para adicionar cores e formas aos elementos HTML, o que não conseguíamos atingir com apenas CSS. Além da redução de arquivos para se trabalhar e a ausência de dependências externas que impactam na performance dos sites, a flexibilidade dessas propriedades permite diversas combinações diferentes, que geram muitos estilos que você não imaginaria ser tão simples de se criar com apenas CSS.

Toda a parte de estilo e adaptação para dispositivos móveis da Website foi feito utilizando o CSS3.



### **4. Análise do sistema a desenvolver**

Neste capítulo irei abordar sobre a análise dos requisitos do meu sistema e a sua modelagem feita através de anotações gráficas através de diagramas da UML.

#### **4.1. Visão geral do Sistema**

Levando em conta os objetivos que foram propostos para o meu sistema, esse será composto da seguinte maneira:

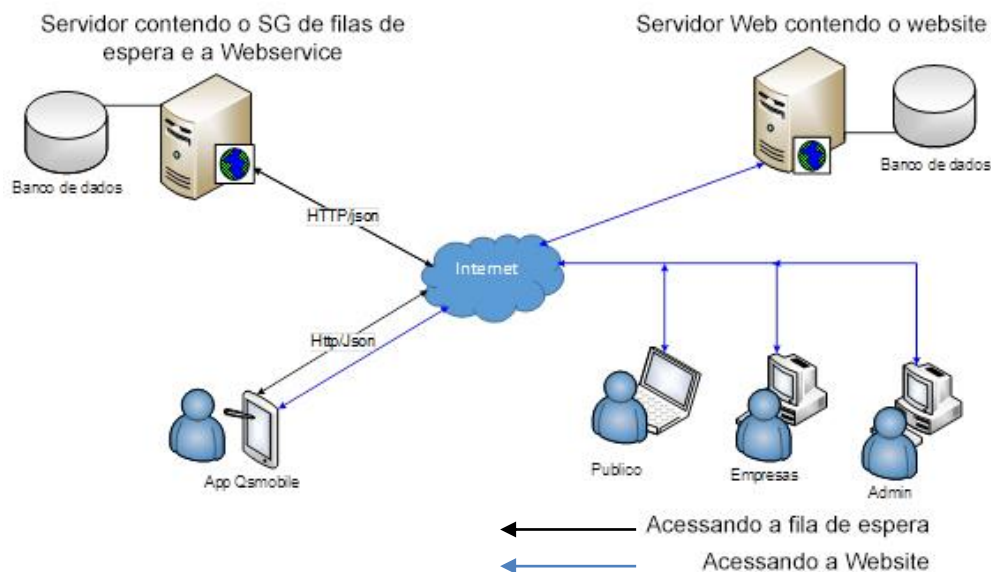
##### **Na parte da aplicação**

Haverá um servidor Web alojando a base de dados MYSQL que conterá informações sobre alguma fila de espera e o Web Services feito em PHP.

Para aplicação conseguir comunicar com a Web Services e obter as informações acerca da fila de espera, será utilizado o protocolo HTTP para solicitação do serviço e a resposta será recebida em formato JSON, que será tratada e mostrada na tela da aplicação adequadamente, mas há ter acesso a internet para isso.

##### **Na parte da Website**

A Website em si estará alojado num servidor web que também terá a sua base de dados MYSQL e poderá ser acedido pela internet através de dispositivos móveis e computadores desktop.



*Figura 12-Arquitetura do sistema proposto*

*Fonte: Elaborado pelo autor*

## 4.2. Análises de Requisitos do Sistema

Os requisitos do sistema estão divididos em, funcionais e não funcionais:

**Requisitos funcionais** - são requisitos do sistema que especificam funções que o sistema ou componente deve ser capaz de realizar, definem o comportamento do sistema, ou seja o processo ou transformação que componentes de software ou hardware efetuam sobre as entradas para gerar as saídas. Esses requisitos capturam as funcionalidades sob o ponto de vista do usuário.

**Requisitos não funcionais** - são os requisitos que não estão diretamente relacionados à funcionalidade do sistema ou seja são as propriedades e características desejadas do sistema relativas à capacidade de armazenamento, tempo de resposta, configuração, uso (ex. uso intuitivo), confiabilidade, entre outros.

### 4.2.1. REQUISITOS FUNCIONAIS

Nesta parte aqui serão mencionados os requisitos funcionais do sistema.

**Para a Aplicação:**

- ***Registrar utilizadores*** - o sistema terá de permitir o cadastro dos utilizadores com nome, email e senha.
- ***Realizar login***- o sistema terá de permitir o login dos utilizadores.
- ***Mostrar lista de empresas disponíveis*** - deverá permitir que o utilizador veja a lista de empresas disponíveis na App.
- ***Apresentar informações das empresas*** - deverá permitir visualizar informações das empresas para que não haja enganos visto que algumas empresas possuem mais que um estabelecimento.
- ***Consultar fila de empresas inseridas na App*** - terá de permitir ao utilizador aceder as filas de espera e visualizar toda a informação correspondente.
- ***Ativar e desativar recebimento de notificações*** - Permitirá ao utilizador ativar o recebimento de notificações em certos intervalos de tempo mesmo quando estiver fora da App, assim não correrá o risco de esquecer de visualizar o andamento das filas. E no final depois de ser atendido poderá desligar as notificações.
- ***Envio de email de reclamações*** - caso tiver algum problema com a App poderá enviar um email comunicando-o a partir da própria App.

#### **Para a Website:**

- ***Realizar cadastro e login de empresas*** - Permitirá empresas cadastrarem e fazerem login para ter acesso à área reservada para empresas.
- ***Mostrar informações úteis da App*** - Terá de mostrar todas as informações da App mesmo que não esteja cadastrado.
- ***Disponibilizar download da App*** – Os utilizadores poderão fazer o download da app mesmo sem estarem cadastrados.
- ***Preencher formulário para requisitar serviço*** - caso a empresa quiser fazer parte da app poderá preencher um formulário e requisitar o serviço.

#### **Painel de administração da Website:**

- **Login administrador** - permitirá o login do administrador.
- **Gerir pedidos** - deverá permitir ao administrador gerir todos os pedidos feitos pelas empresas.
- **Gerir utilizadores** - permitirá gerir os utilizadores da Website.
- **Enviar confirmação por email** - ao receber os pedidos o administrador poderá confirma-los enviando um email para o cliente à partir do painel de administração.
- **Configurações**- permitirá a um administrador cadastrar outros administradores.

#### 4.2.2. REQUISITOS NÃO FUNCIONAIS

Visto os requisitos funcionais agora veremos os requisitos não-funcionais do sistema desta pesquisa, agrupados pela sua classificação.

**Usabilidade** – O sistema deverá conter uma boa aparência visual, com tudo bem estruturado e simplificado. A resolução mínima que irá permitir é de 240 X 320 e poderá funcionar normalmente com a tela invertida (horizontal e vertical). Funcionará normalmente em algumas partes sem precisar de internet, mas quando chegar na parte de consultar as filas ali já não irá funcionar, será notificado ao utilizador uma mensagem de erro e o utilizador deverá ligar a internet.

A interface da Web Site deverá se comportar adequadamente independente da forma como será acessado, seja Browser, Smartphone ou Tablet. Ou seja deverá possuir um design responsivo, com informação bem estruturadas de forma que os utilizadores as possam encontrar facilmente.

**Operacionais** – terá de operar na plataforma de android 4.0 ou superior e é obrigatório o uso de internet para ter acesso as filas de espera.

A Website terá que estar hospedada num servidor Web logicamente para ser acessado pelo público.

**Implementação** – A parte da App será desenvolvida na linguagem java, o módulo servidor utilizará a linguagem PHP e o banco de dados utilizará a linguagem SQL. Na parte web será desenvolvida com linguagem HTML, PHP e CSS.

**Portabilidade** – A App poderá ser instalado em qualquer dispositivo que funcione com a plataforma android. E a Website não necessita de um servidor que possua características muito elevadas mas sim só o necessário.

**Facilidade e confiabilidade** – Tanto a Website ou a App terão de ter um layout de fácil compreensão por parte dos utilizadores e terão de transmitir credibilidade para com esses.

**Arquitetura** - Para que a camada servidora opere normalmente será utilizado o Windows como sistema operacional, o servidor terá que ter no mínimo 512 megabytes de memória RAM com capacidade para receber futuros upgrates tendo em conta o conceito de a escalabilidade ou seja o aumento significativo do número de utilizadores. Para a parte cliente utilizará o sistema operacional Android. O MYSQL será o SGBD utilizado para o gerenciamento dos dados que juntamente com a WebServices e a Website serão suportados pelo servidor apache. Toda a comunicação entre o cliente e o servidor será feita com o WebService REST utilizando protocolos como HTTP e as mensagens terão o formato de JSON.

### 4.3. Desenho do Sistema

Para fazer a modelação do Sistema foi utilizado uma notação gráfica standard - UML - que irá modelar o sistema independentemente da linguagem de programação utilizada.

Segundo Alberto da Silva & Carlos Videira (UML, Metodologias e Ferramentas CASE, 2001) o UML (*Unified Modelling Language*) é uma linguagem diagramática, utilizável para especificação, visualização e documentação de sistemas de software. O UML surgiu em 1997 na sequência de um esforço de unificação de três das principais linguagens de modelação orientadas por objetos (OMT, Booch e OOSE).

A UML disponibiliza um conjunto de diferentes tipos de diagramas, dentre as quais serão mostrados:

- Diagrama de Use Cases;
- Diagrama de Classes;
- Diagrama de Sequência;
- Diagrama de Atividade;

#### 4.3.1. DIAGRAMAS DE CASO DE USO

Segundo Alberto da Silva & Carlos Videira (UML, Metodologias e Ferramentas CASE, 2001, Pag.145) um diagrama de use case ou caso de utilização é uma sequência de ações que um ou mais atores realizam num sistema de modo a obterem um resultado particular.

O modelo de casos de utilização permite capturar os requisitos de um sistema através do detalhe de todos os cenários que os utilizadores podem realizar. Os casos de utilização, mais que iniciar a modelação de requisitos de um sistema, dirigem/conduzem todo o processo de desenvolvimento.

Na Figura 13, é apresentado o diagrama de caso de uso resultante do processo de panejamento do projeto, na visão do autor para o utilizador da App.

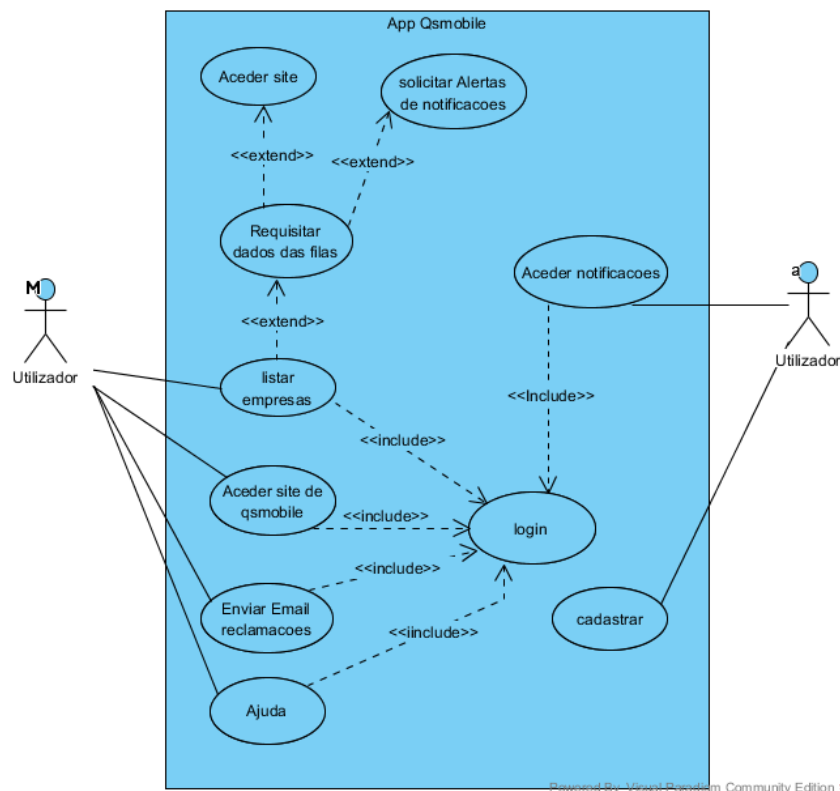


Figura 13-Diagrama de Use Case App Qsmobile

Fonte: elaborado pelo próprio autor

E na figura 14 temos o diagrama de use case para utilizadores da Website.

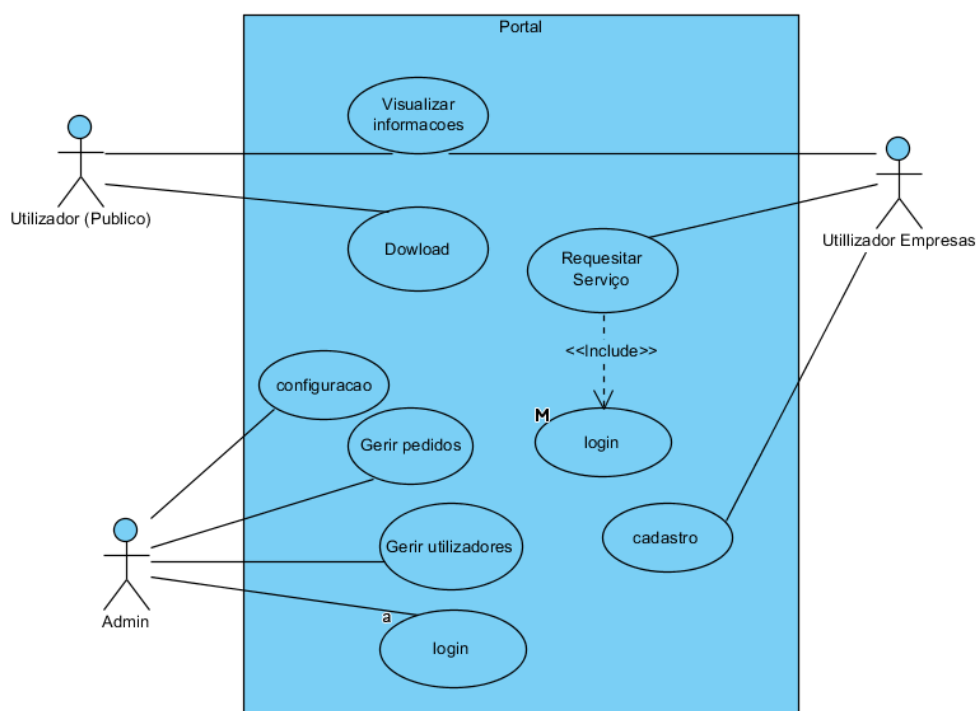


Figura 14- Diagrama de use case para utilizadores da Website

Fonte: Elaborado pelo próprio autor

#### 4.3.2. DIAGRAMA DE ATIVIDADE

Segundo Alberto da Silva & Carlos Videira (UML, Metodologias e Ferramentas CASE, 2001, Pag.222) um diagrama de atividades é um caso particular de um diagrama de estados, no qual todos ou a maioria dos estados são “estados de atividades” e todas ou a maioria das transições são desencadeadas pela conclusão das atividades dos estados anteriores.

Ainda segundo esses os diagramas de atividades correspondem aos conhecidos fluxogramas. Fornecem uma visão simplificada do fluxo de controlo de uma operação ou de um processo de negócio, também designado por “*workflow*”.

A figura 15 a baixo descreve o diagrama de atividades da App android Qsmobile.

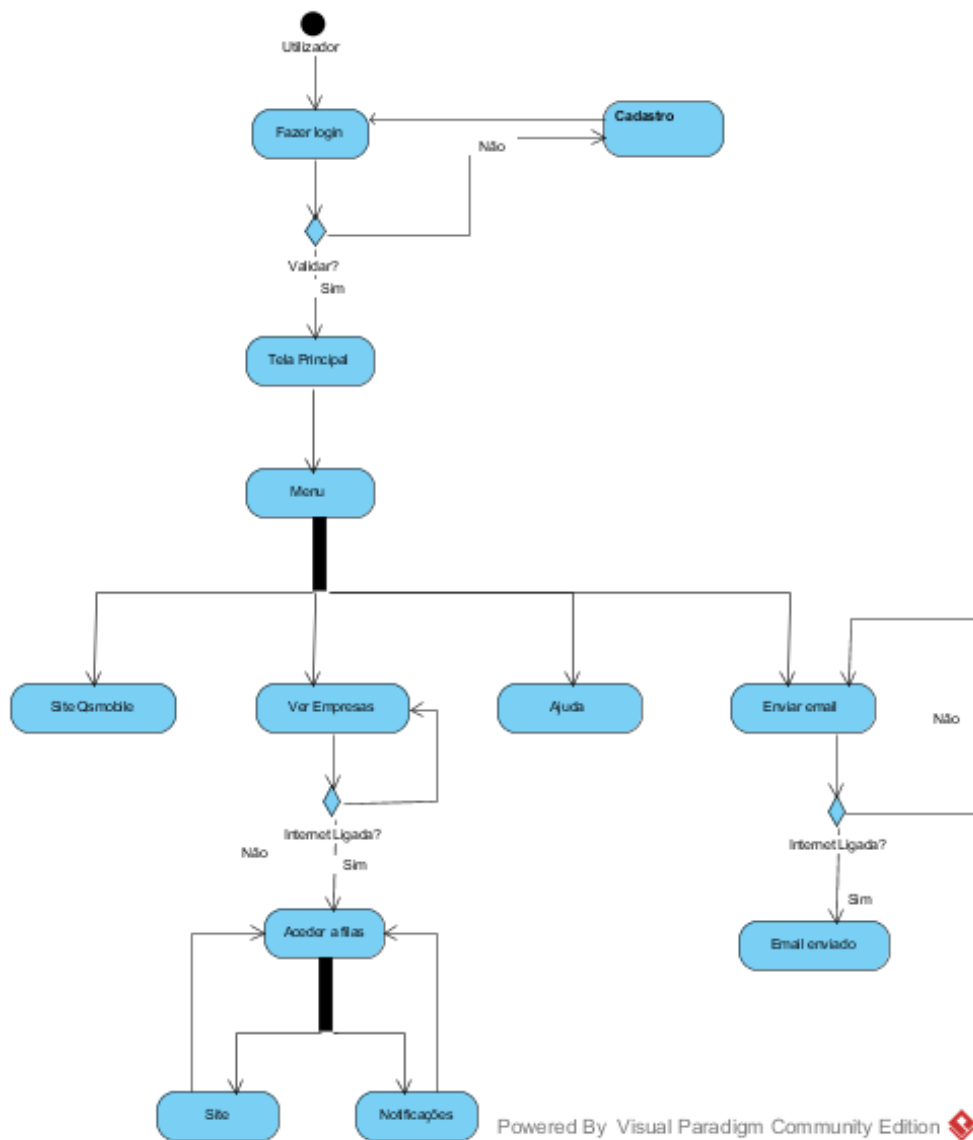


Figura 15-Diagrama de Atividades da App QSmobile

Fonte: Elaborado pelo próprio autor

#### 4.3.3. DIAGRAMA DE CLASSES

Segundo Alberto da Silva & Carlos Videira (UML, Metodologias e Ferramentas CASE, 2001, Pag.186) um diagrama de classes ilustra um conjunto de classes, interfaces, colaborações e respetivas relações, em geral de dependência, generalização e de associação.

Os diagramas de classes são usados para modelar a estrutura de um sistema. Estes modelos são também designados por “vista do desenho estático do sistema” e são



usados tipicamente em três situações: (1) para modelar o vocabulário de um sistema; (2) para modelar colaborações simples; e (3) para modelar o desenho de um esquema de uma base de dados.

Na figura 16 descreve o diagrama de classes da App android.

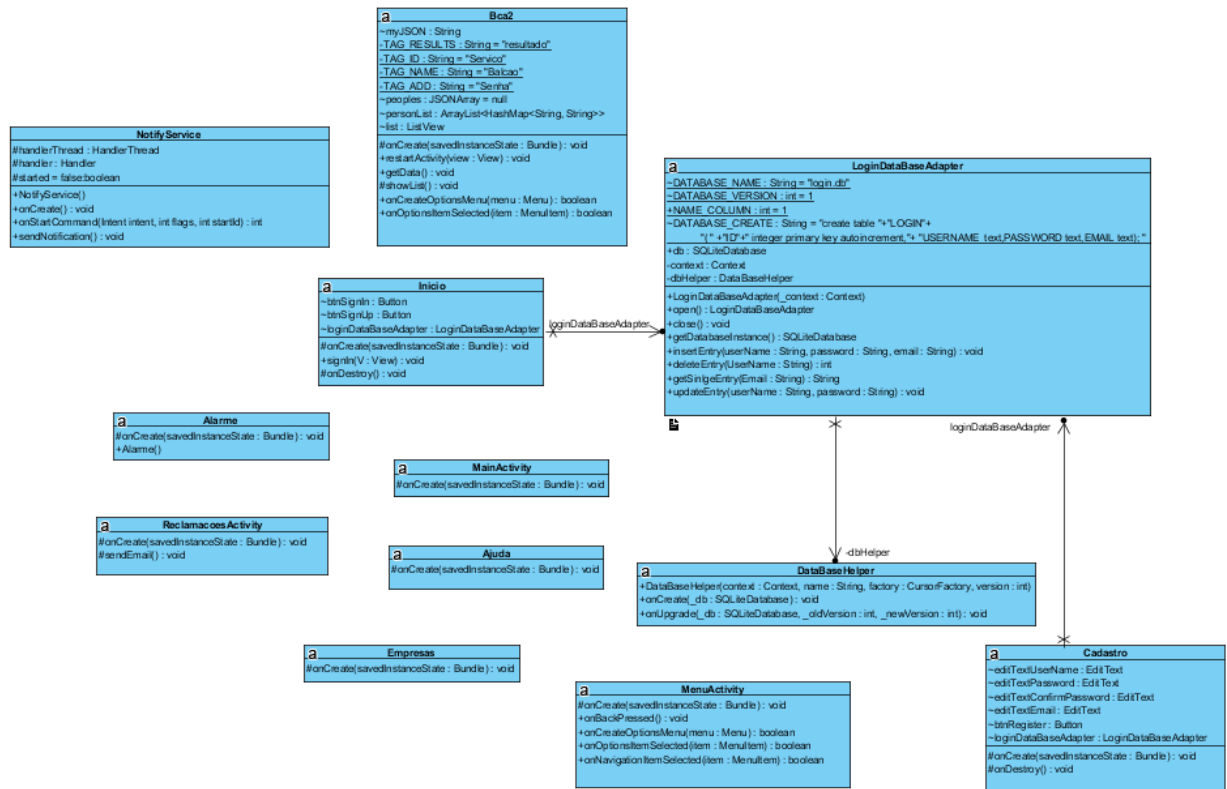


Figura 16-Diagrama de classe da App android

Fonte: Elaborado pelo próprio autor

#### 4.3.4. DIAGRAMA DE SEQUÊNCIA

Segundo Alberto da Silva & Carlos Videira (UML, Metodologias e Ferramentas CASE, 2001, Pag.204) Um diagrama de sequência ilustra uma interação segundo uma visão temporal. Um diagrama de sequência é representado através de duas dimensões: a dimensão horizontal, que representa o conjunto de objetos intervenientes; e a dimensão vertical que representa o tempo.

Nos diagramas de sequência as setas são desenhadas horizontalmente de forma a representar a indivisibilidade da operação necessária para enviar o estímulo. Esta assunção é válida para a generalidade das situações.

Na figura 17 temos o diagrama de sequência para caso de uso requisitar dados das filas.

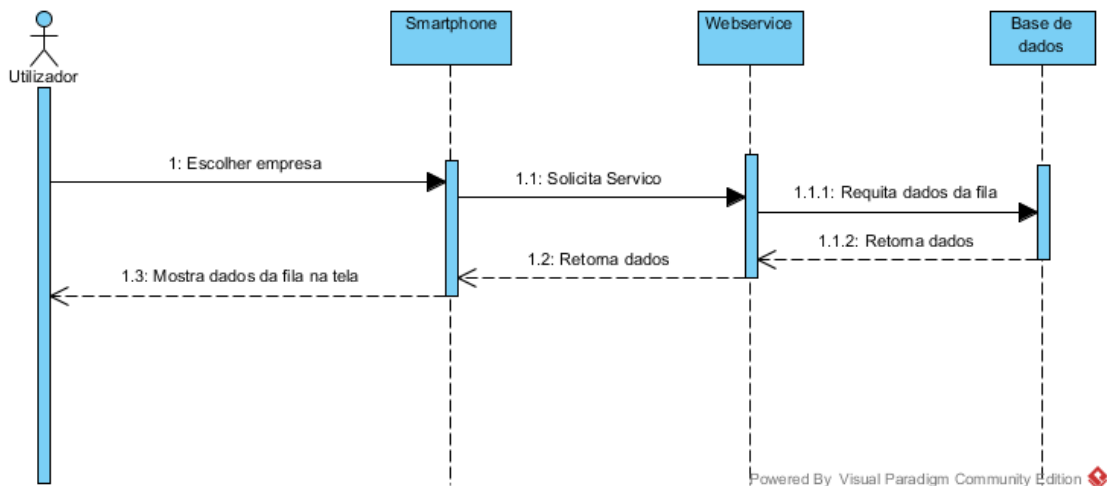


Figura 17-Diagrama de sequência para caso de uso requisitar dados das filas

Fonte: Elaborado pelo próprio autor

Na figura 18 temos outro diagrama de sequência mas agora para solicitar alertas por notificações.

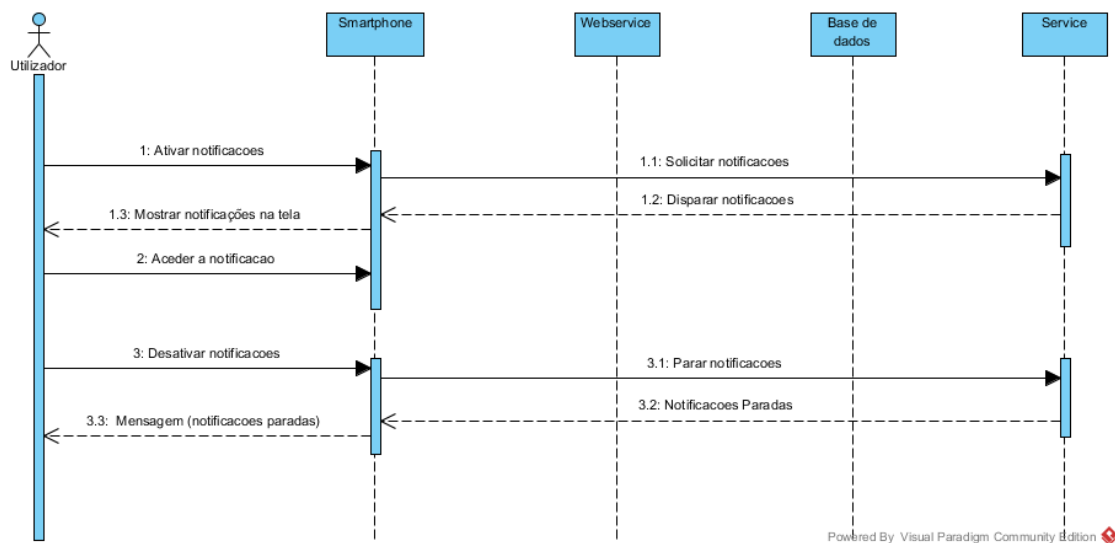


Figura 18-Diagrama de sequência de caso de uso solicitar alertas por notificação

#### 4.3.5. BASE DE DADOS

Na parte de armazenamento de dados de toda a parte da Website foi utilizado o SGBD MYSQL para criar toda a estrutura do banco de dados.

A figura abaixo demonstra as tabelas da base de dados da Website.

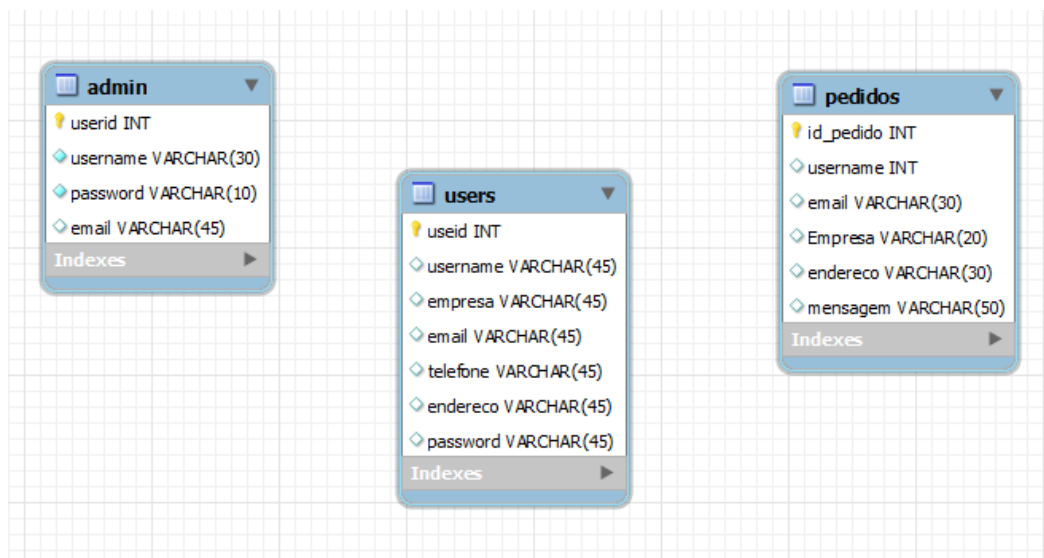


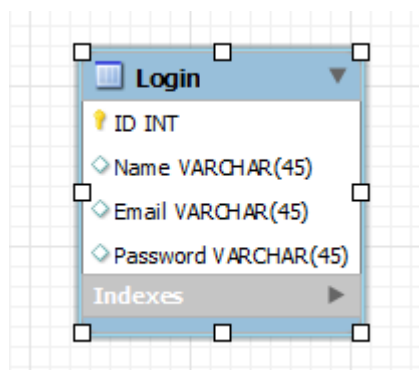
Figura 19-Tabelas da base de dados da Website

Fonte: Elaborado pelo próprio autor

E também esse mesmo SGBD foi utilizado na parte de consultas dos dados das filas de espera. Foi criado um simulador que inclui uma BD MYSQL, e esse consegue manipular os dados da tabela chamando uma nova senha e disponibiliza-lo para que o utilizador final possa visualiza-lo como se fosse o andamento de uma fila de espera em tempo real. Então A aplicação consegue aceder a esses dados através do Web service.

Para o armazenamento dos dados do utilizadores da aplicação android foi utilizado uma base de dados local que e o SQLite, não tendo necessidade de ter uma base de dados no externa.

Na figura a seguir podemos ver a estrutura da base de dados da aplicação.



*Figura 20-Tabela da base de dados da aplicação android*

*Fonte: Feita pelo proprio autor*

### 5. Implementação do Protótipo

Segundo Robson Santos (2007), protótipos são modelos construídos para simular a aparência e a funcionalidade de um produto em desenvolvimento. Um protótipo é uma representação da interface com a qual o usuário pode interagir e oferece informações para propor mudanças e melhorias.

Um protótipo pode ser:

- um pequeno guia para o usuário;
- uma simulação em papel;
- uma simulação usando o próprio computador a partir de uma ferramenta de prototipagem;
- uma versão inicial do programa;
- o sistema a ser substituído, em funcionamento.

Por meio de um protótipo, os futuros utilizadores do *software* e desenvolvedores, podem interagir, avaliar, alterar e aprovar as características mais marcantes da interface e da funcionalidade da aplicação.

Quando se desenvolve um protótipo não é necessário representar todas as funcionalidades do produto. Pelo contrário, é necessário representar apenas as funcionalidades que atendam aos objetivos do teste ou da avaliação.

#### 5.1. Configuração da aplicação

Na plataforma android há um arquivo chamado de `AndroidManifest.xml` que se encontra na pasta raiz do projeto, é obrigatório o seu uso porque nele ficam configurados todos os recursos utilizados pela aplicação, recursos como atividades, permissões, services entre outros.

Na figura a seguir mostra o arquivo `AndroidManifest.xml` da aplicação QSmobile juntamente com todas as suas configurações.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.strok.mobileapp" >

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label=""
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label=""
            android:theme="@style/AppTheme.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Inicio" >
        </activity>

        <activity
            android:name=".MenuActivity"
            android:label="MenuActivity"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity
            android:name=".empresas"
            android:label="empresas"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity
            android:name=".Novobanco"
            android:label="Novobanco"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity
            android:name=".Novobanco"
            android:label="Novobanco"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity android:name=".Ajuda" >
        </activity>
        <activity android:name=".alarme" >
        </activity>
        <activity
            android:name=".Bca2"
            android:label="Bca2"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>

        <service
            android:name=".NotifyService"
            android:exported="false" />

        <activity
            android:name=".ReclamacoesActivity"
            android:label="ReclamacoesActivity"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
        <activity
            android:name=".Cadastro"
            android:label="Cadastro"
            android:theme="@style/AppTheme.NoActionBar" >
        </activity>
    </application>
</manifest>

```

Figura 21- Configuração do arquivo AndroidManifest.xml

## 5.2. Criação da base de dados local (SQLite)

Para criar a base de dados local da aplicação primeiro tive que se criar uma classe DataBaseHelper que estende a classe SQLiteOpenHelper que é responsável por gerenciar todas as operações relacionadas com o banco de dados. A figura seguinte um trecho do código da classe.

```
public class DataBaseHelper extends SQLiteOpenHelper
{
    public DataBaseHelper(Context context, String name, CursorFactory factory, int version)
    {
        super(context, name, factory, version);
    }
}
```

Figura 22-Classe DataBaseHelper

Depois criei a classe LoginDataBaseHelper que irá criar a base de dados e todas as tabelas necessárias. Também nela contem os métodos de inserção, atualização, leitura e dos dados. Essa classe está interligada com a classe DataBaseHelper que foi mencionado anteriormente. Confira a figura a baixo.

```
public class LoginDataBaseAdapter
{
    static final String DATABASE_NAME = "login.db";

    static final int DATABASE_VERSION = 1;

    public static final int NAME_COLUMN = 1;
    // TODO: Crie um campo público para cada coluna em sua tabela.
    // SQL Statement para criar a base de dados.
    static final String DATABASE_CREATE = "create table "+ "LOGIN"+
        "( " + "ID"+" integer primary key autoincrement,"+ "USERNAME text,PASSWORD text,EMAIL text); ";

    // Variável para manter a instância do banco de dados
    public SQLiteDatabase db;
    // Contexto do aplicativo usando o banco de dados.
    private final Context context;
    // Assistente de atualização / atualização de banco de dados
    private DataBaseHelper dbHelper;
```

Figura 23- Constituição da classe LoginDataBaseHelper

## 5.3. Acessando o Webservice

Para fazer a leitura dos dados da fila de espera com a App android, foi usado um Webservice utilizando os protocolos HTTP, GET e formato JSON para receção dos dados.

```

public void getData(){
    class GetDataJSON extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {

            BufferedReader inputStream = null;

            String result = null;
            try {

                URL url = new URL("http://10.127.127.1/ws/tex.php");
                HttpURLConnection con = (HttpURLConnection) url.openConnection();
                StringBuilder sb = new StringBuilder();
                inputStream = new BufferedReader(new InputStreamReader(con.getInputStream()))
                String line = null;
                while ((line = inputStream.readLine()) != null)
                {

```

*Figura 24- Método para fazer leitura e disponibilização dos dados*

## 5.4. Obter dados das filas

A requisitar os dados de uma fila é nos retornados alguns dados como mostra a seguinte figura:

```

{"resultado":[{"Servico":"Prioridade","Balcao":"C","Senha":"1"}, {"Servico":"Ate.Geral","Balcao":"B","Senha":"0"}, {"Servico":"Ate.Caixa","Balcao":"A","Senha":"3"}, {"Servico":"Empresas","Balcao":"D","Senha":"0"}]}

```

*Figura 25-Dados de uma fila de espera no json*

A aplicação android ao receber esses dados ele irá trata-los e apresenta-los na tela adequadamente para que o utilizador possa visualiza-los corretamente. O layout foi desenvolvido com a finalidade de disponibilizar os dados de uma forma clara e objetiva.

## 5.5. Service (Notification Manager)

Para que seja possível receber notificações tive que utilizar services como já tinha mencionado anteriormente. A service criada funciona de forma simples, ela irá disparar notificações em certos intervalos de tempo mesmo que o utilizador estiver com a App em background ou seja fechada. Para as notificações entrarem e preciso ativa-las e quando já não precisar mais delas será necessário desativa-las se não elas continuaram a entrar constantemente consumindo assim energia do seu telemóvel. Ao clicar na notificação recebida ela irá leva-lo diretamente para consultar a fila. A figura a baixo mostra um trecho do código.



```

public class NotifyService extends Service {

    private HandlerThread handlerThread;
    private Handler handler;
    private boolean started = false;

    //Define o tempo entre notificações, altere como quiser
    private final int TEMPO_ENTRE_NOTIFICAÇÕES_SEGUNDOS = 7;

    @Override
    public void onCreate() {
        Log.d("NotifyService", "onCreate");

        handlerThread = new HandlerThread("HandlerThread");
        handlerThread.start();
        handler = new Handler(handlerThread.getLooper());
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        Log.d("NotifyService", "onStart");

        if(!started) {
            Log.d("NotifyService", "Notificações iniciadas");
            started = true;
            Runnable runnable = new Runnable() {

```

*Figura 26- Código para disparar notificações*

## 5.6. Envio de Email (comunicar problemas)

A parte de envio de emails serve para comunicar problemas a cerca da App. Para conseguir isso tive que usar a ação ACTION\_SEND para lançar um cliente de email instalado no seu dispositivo Android. Para que haja o envio de emails é preciso utilizar o emailIntent. A seguir veja a sintaxe para criar uma intenção com a ação ACTION\_SEND.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
```

A figura abaixo representa um trecho do código para envio de emails:

```

protected void sendEmail() {
    Log.i("Send email", "");
    String[] TO = {" "};
    String[] CC = {" "};
    Intent emailIntent = new Intent(Intent.ACTION_SEND);

    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.setType("text/plain");
    emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
    emailIntent.putExtra(Intent.EXTRA_CC, CC);
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Assunto");
    emailIntent.putExtra(Intent.EXTRA_TEXT, "Mensagem");

    try {
        startActivity(Intent.createChooser(emailIntent, "Enviando o seu mail..."));
        finish();
        Log.i("Finished sending email...", "");
    }
}

```

*Figura 27- Envio de emails*

## 5.7. Adaptação da Website para dispositivos móveis

O volume de usuários que acessam a Internet por meio de dispositivos móveis cresceu exponencialmente nos últimos anos, tendo em conta esses aspectos então faria mais sentido que a minha Website funcionasse para dispositivos móveis também e não só para computadores desktop. Para que isso seja possível ela deve se adaptar a diversos tamanhos como exemplo, Smartphone, Tablet ou outro qualquer. Para resolver isso utilizei a linguagem CCS, ao criar uma parte de estilo para o Website no CSS deve-se criar outra vez essa parte mas agora para fazer a adaptação e isso tudo foi feito dentro do mesmo Script CSS.

A figura abaixo mostra o trecho de código CSS de uma adaptação feita para uma tela com 980px.

```

* Containers */

.container {
    margin-left: auto;
    margin-right: auto;
}

.container.\31 25\25 {
    width: 100%;
    max-width: 87.5em;
    min-width: 70em;
}

.container.\37 5\25 {
    width: 52.5em;
}

.container.\35 0\25 {
    width: 35em;
}

.container.\32 5\25 {
    width: 17.5em;
}

.container {
    width: 70em;
}

@media screen and (max-width: 980px) {

    .container.\31 25\25 {
        width: 100%;
        max-width: 125%;
        min-width: 100%;
    }

    .container.\37 5\25 {
        width: 75%;
    }

    .container.\35 0\25 {
        width: 50%;
    }

    .container.\32 5\25 {
        width: 25%;
    }

    .container {
        width: 100% !important;
    }

}

```

*Figura 28-Adaptação para dispositivos móveis tela 980px*

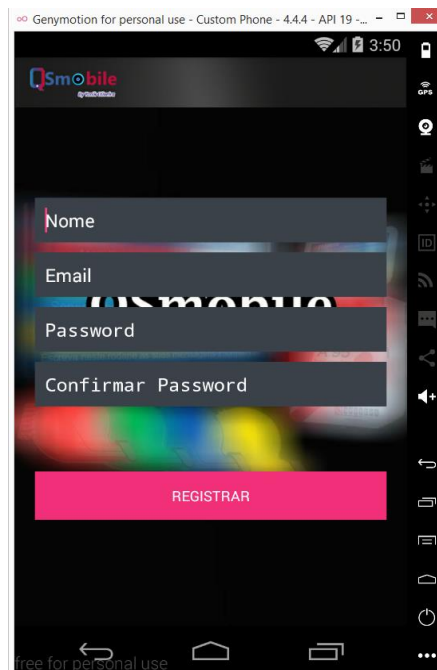
No lado esquerdo temos a parte somente de estilo e na direita a parte com estilo e adaptação feita. Esse trecho faz com que o conteúdo da página Web fique centrado atribuindo algum espaço nas margens da esquerda e da direita.

## 5.8. Funcionamento do projeto

Neste ponto irei mostrar o funcionamento de todo o projeto, mostrarei as telas da aplicação já executando e explicarei o que cada uma faz. A interface foi desenvolvida com ênfase na fácil compreensão por parte dos utilizadores que já costumam interagir com dispositivos móveis e as suas aplicações. Também mostrarei o funcionamento da parte web que também é de fácil compreensão para utilizadores que já costumam utilizar Websites.

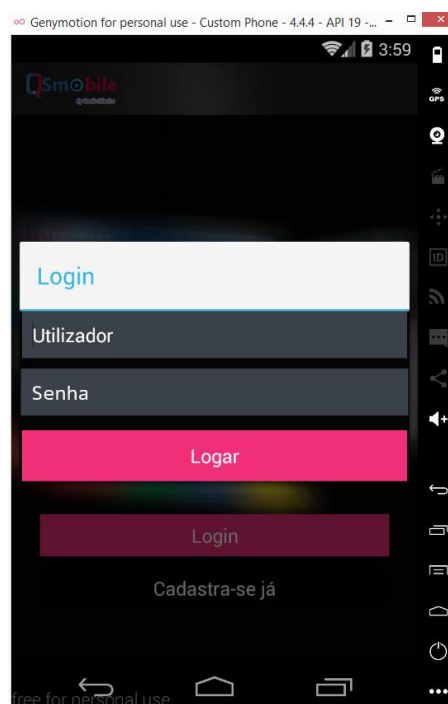
### 5.8.1. APLICAÇÃO QSMOBILE

Ao iniciar a aplicação pela primeira vez o utilizador precisará se cadastrar para ter acesso ao conteúdo da aplicação. Primeiramente não é preciso internet para essa tarefa, a figura a seguir mostra o formulário de cadastro:



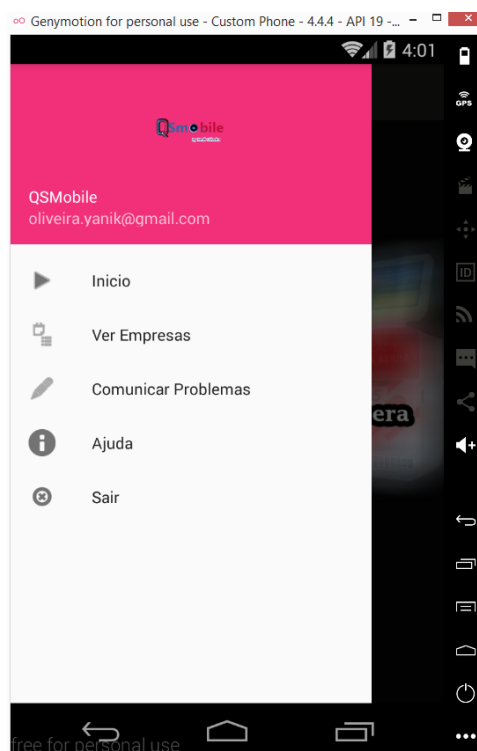
*Figura 29-Formulário para cadastro do utilizador*

Feito isso, se tudo correr bem o utilizador ficará cadastrado na base de dados e já poderá ter acesso aos recursos fornecidos pela aplicação. A figura seguinte mostra a parte de login:



*Figura 30-Formulario de login*

Feito o login o utilizador poderá ver a tela principal da aplicação que possui um menu deslizante com as seguintes opções como mostra a figura a baixo.



*Figura 31-Menu da aplicação*

Ao clicar na opção “Ver Empresas” o utilizador poderá ver a lista das empresas ou instituições disponíveis na app com os respetivos endereços para que o utilizador não confunda as empresas, veja a figura 32.



*Figura 32-Lista de empresas disponíveis*

Já nessa fase se o utilizador quiser consultar os dados das filas terá que ligar a internet caso contrario não funciona e mostra na tela mensagem de erro. Com a internet ligada ao clicar numa das empresas da lista vai conseguir ver os dados da fila de espera da empresa escolhida. A figura seguinte mostra os dados da fila de espera do banco BCA, que foi utilizado como exemplo na aplicação.

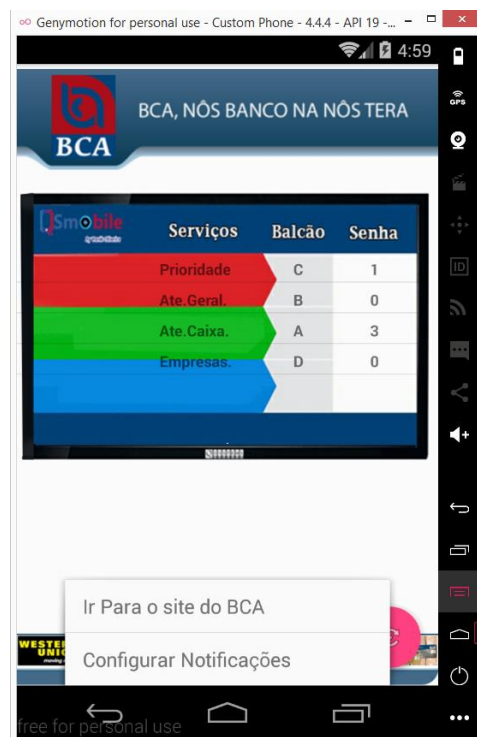
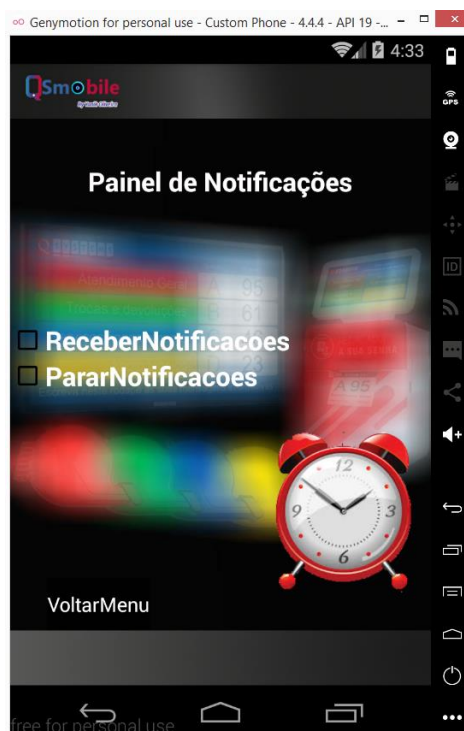


Figura 33-Layout de Fila de espera Ex: BCA

Também nesse layout tem um menu com as opções como mostra a figura 33, ao clicar em “Ir Para o site do BCA” a opção fala por si, é redirecionado para o site da empresa, na outra opção pode-se ativar e desativar as notificações no painel de configuração das notificações. Como mostra a figura 34.



*Figura 34-Painel de notificações*

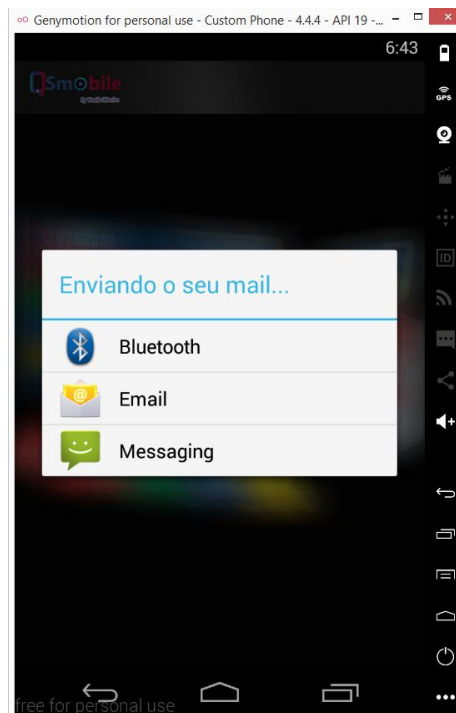
Escolhendo a opção “ReceberNotificoes” receberá notificações no seu dispositivo em certos intervalos de tempo mesmo que a app esteja fechada. Quando a notificação entrar é só clicar nela que ele redirecionará o utilizador para a fila de espera que tinha sido requisitada. Depois é só desligar as notificações para que essas possam parar de entrar no dispositivo. Há que ter em conta que se receber uma notificação e essa não for aberta ou eliminada o utilizador não recebera mais notificações. Na figura a seguir podemos ver uma notificação que entrou mesmo a aplicação estando fechada.



*Figura 35- Exemplo de notificação entrando*

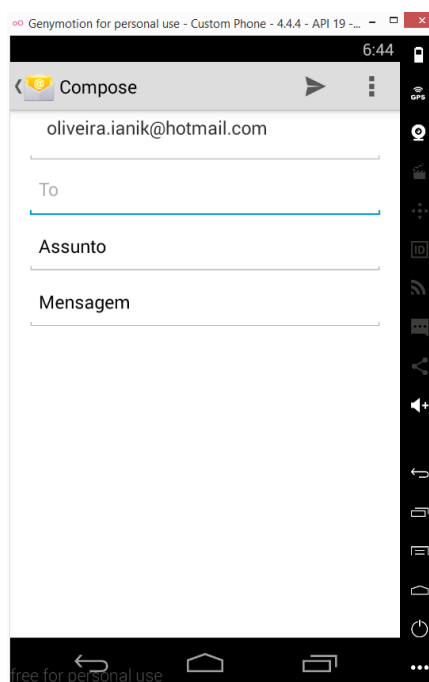
Voltando ao menu temos outras opções como “Comunicar problemas” que funciona com envio de emails a partir de clientes que estão instalados no telefone ou seja aplicações de emails como Gmail, Outlook ou outro qualquer.





*Figura 36-Conectando com cliente*

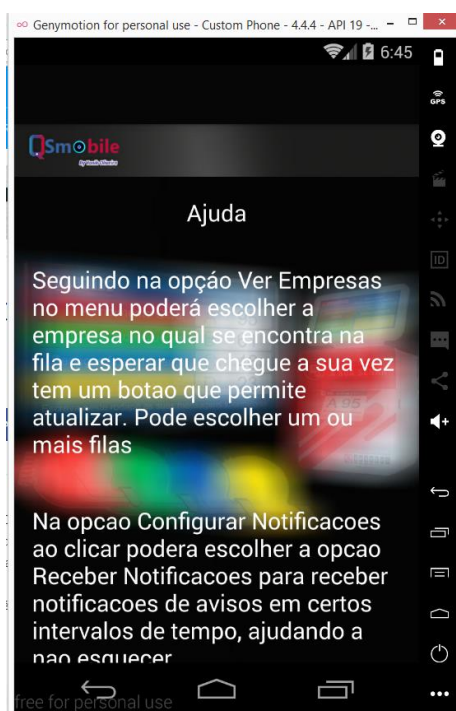
No meu caso tenho instalado apenas o Outlook então é a partir desse que irei enviar os meus emails ma se caso tivesse outros também funcionaria em pleno. Na figura seguinte podemos ver a composição de emails.



*Figura 37-Composição de um email*

E só preencher com os dados necessários que o envio é feito normalmente para o email caso inserido corretamente.

No menu inicial também possui a opção ajuda que é muito encontrado em vários tipos aplicações, nela podem se esclarecer algumas dúvidas e aprender a manusear a aplicação QSmobile caso o utilizador não esteja conseguindo utiliza-lo corretamente. A seguinte figura 38 mostra a tela de ajuda.



*Figura 38- Tela de ajuda para utilizadores*

### 5.8.2. WEBSITE

Trata-se de um Website simples que tem como objetivo disponibilizar informações e o download da aplicação para a parte pública e também possui a parte restrita para empresas onde essas podem entrar em contato e solicitar esse serviço e serem inseridos na app.

A figura a seguir mostra a parte inicial pública onde tem várias informações acerca da app e também pode-se ver o menu que é bastante simples e claro.

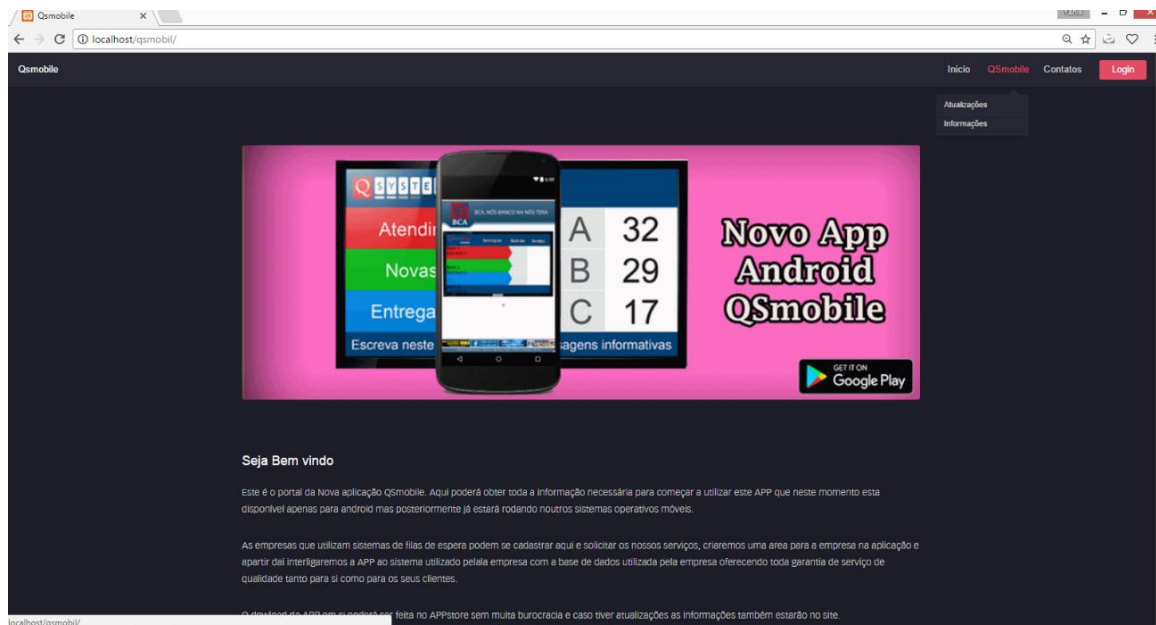


Figura 39-Parte inicial para público

Podemos ver a área de cadastro para empresas como mostra a figura seguinte.

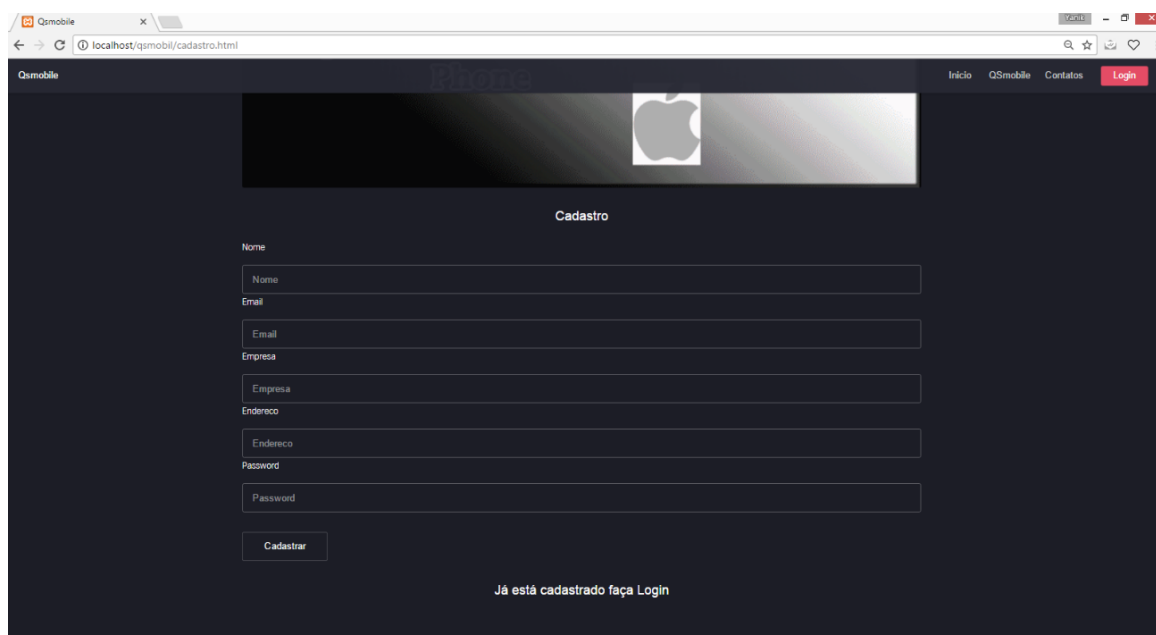
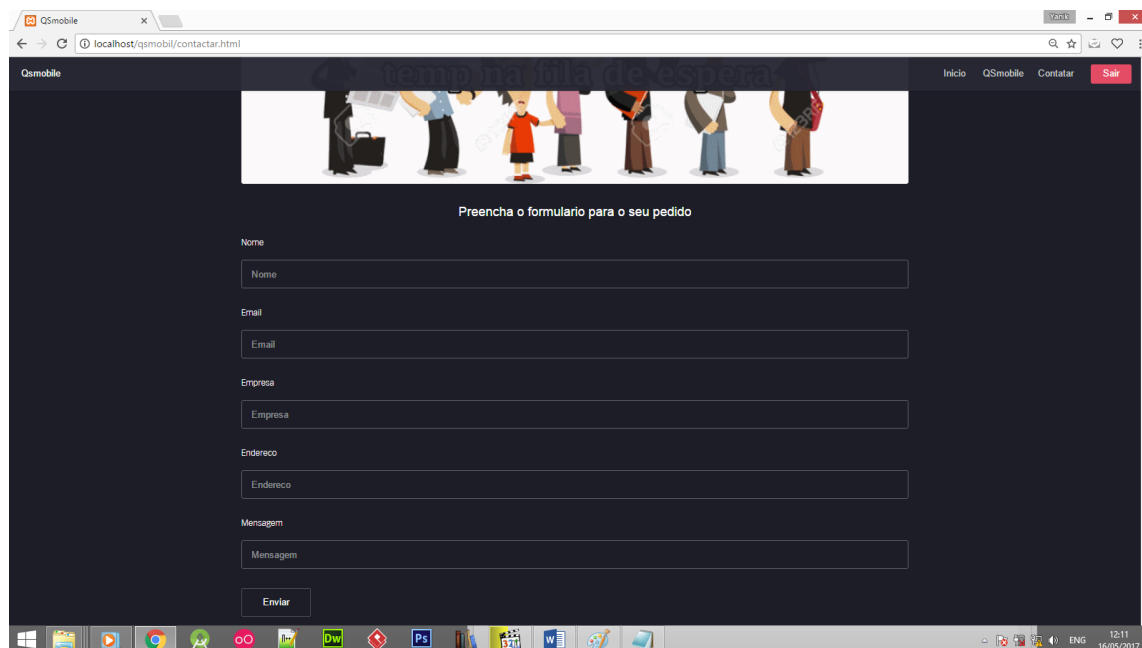


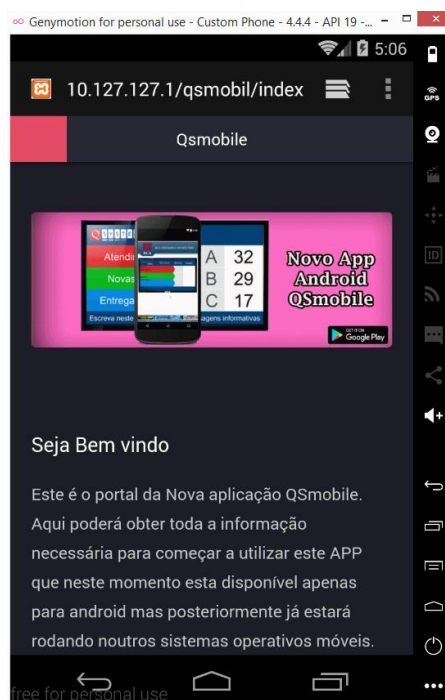
Figura 40- Cadastro de empresas

Ao se cadastrar então a empresa já poderá efetuar o login e entrar em contacto comigo, requisitando os serviços da app QSmobile.



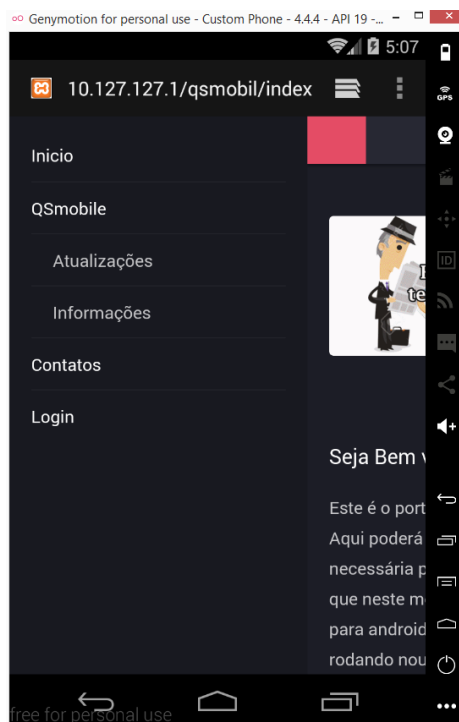
*Figura 41-Formulario de contato para requisitar o serviço*

O Website também estará disponível para dispositivos móveis visto que atualmente muitas pessoas acedem aos sites através dos seus smartphones ou tables e não somente nos seus computadores, a figura abaixo mostra a parte inicial da Website mas agora num dispositivo com android.



*Figura 42-Pagina inicial do Website num dispositivo móvel*

Na figura 43 a seguir, podemos ver a estrutura do menu num dispositivo móvel.



*Figura 43- Menu da Website no android*

Toda a parte web é gerida por um administrador cujo principal objetivo é gerir os utilizadores e os pedidos, a confirmação dos pedidos é feito através de emails a partir do painel administrativo. A figura seguinte mostra o painel do administrador que claramente tem que fazer login para aderir ao painel.

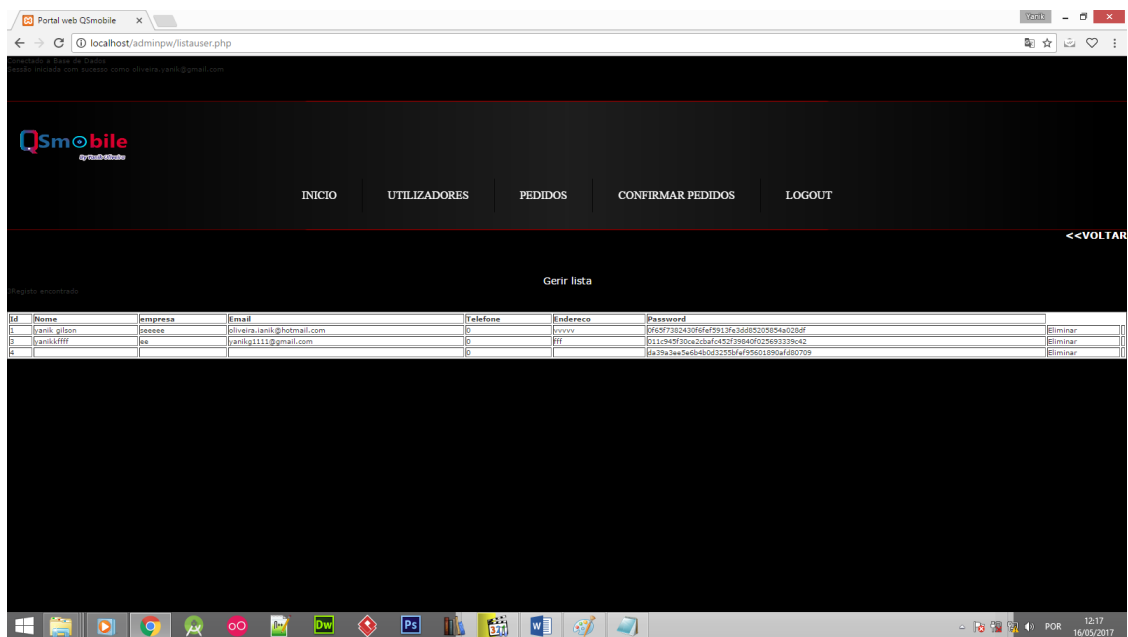


Figura 44-Painel do administrador

Quando se requisita o serviço é aconselhado a inserir o seu email corretamente, porque posteriormente a confirmação desse será redirecionado para esse mesmo email.

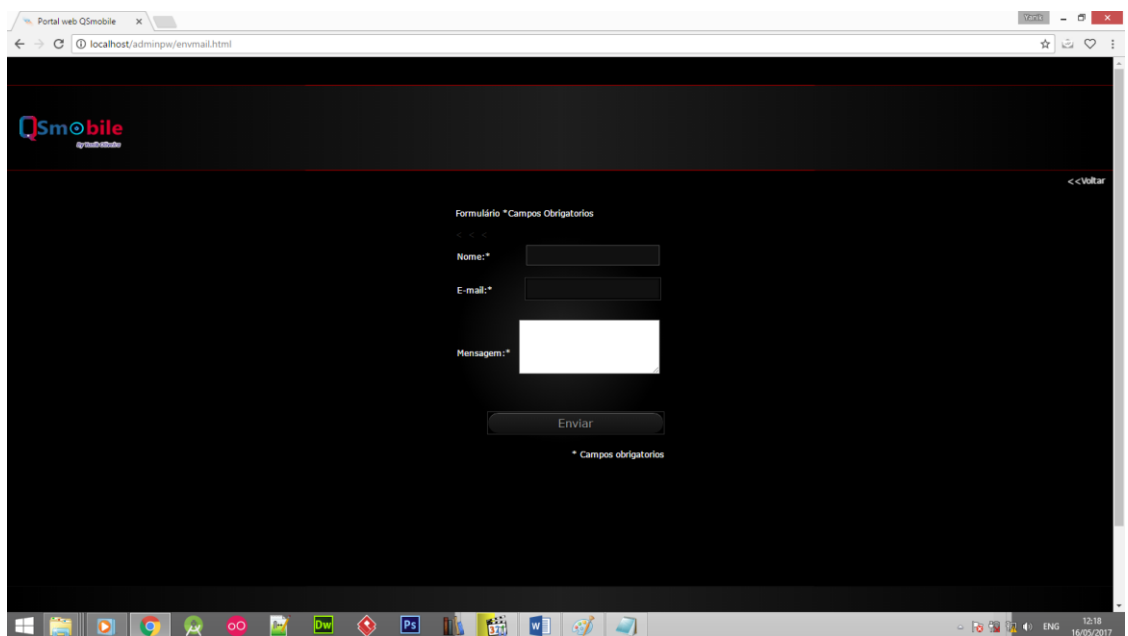


Figura 45- Confirmação dos pedidos através de email

### **6. Conclusão**

#### **6.1. Resultados obtidos**

A implementação desse projeto trará consigo várias vantagens para quem se encontra numa fila de espera graças a ele um cliente ao chegar num estabelecimento e retirar a sua senha caso encontrar uma fila enorme, com muitas pessoas para serem atendidas na sua frente, já não terá problemas em sair do estabelecimento e ir para outro lugar resolver outro assunto. Com essa aplicação o cliente poderá controlar a sua vez de qualquer lugar desde que esse tenha acesso a internet. Com isso o utilizador cliente já não precisa ficar ali na fila esperando horas e mais horas para ser atendido, e quando sair do estabelecimento correr o risco de perder a sua vez. Ele poderá sair e só voltar quando estiver quase chegando na sua vez.

A aplicação também possui alertas por notificação que servem para que o cliente não se esquece de consultar a fila no seu telemóvel. A pessoa ao ir fazer outra coisa por exemplo ir trabalhar, desde de que ative as notificações ele poderá receber notificações em certos intervalos de tempo mesmo que a aplicação estiver fechada, ajudando-o assim a não esquecer de consultar a fila e com isso minimizar o risco de esquecer e perder a sua vez.

A Qsmobile pretende resolver os problemas ou limitações dos sistemas de gestão de filas de espera no que diz respeito ao tempo de espera dos clientes nas filas, e com isso tentar diminuir vários constrangimentos que tem ocorrido nas filas dentro dos estabelecimentos.

#### **6.2. Trabalhos futuros**

Mas o projeto em si não para por aqui, futuramente pretendo que essa tecnologia rode em outros sistemas operacionais móveis como Windows Phone e IOS.

Também será inserido na aplicação a opção de solicitar uma notificação e ser notificado só quando o utilizador quiser, por exemplo imaginamos que eu tenha uma

senha com o número 50, posso ir na aplicação e configurar para receber uma notificação quando estiver na senha 45 por exemplo, assim já não receberá muitas notificações como atualmente.

Pretendo futuramente inserir a opção de retirar senhas virtuais a partir da aplicação desde que o dispensador de senhas interage com uma base de dados, visto que algumas empresas em cabo verde utilizam senhas em rolo o que torna impossível a comunicação com a aplicação mas caso utilize uma base de dados poderá haver comunicação entre esse e a aplicação. Pretendo com isso que um utilizador ao obter a senha com número 10 na aplicação por exemplo, o cliente que for retirar no dispensador de senhas será disponibilizado para ele a senha 11. Há que ter em conta o risco de um utilizador retirar 10 na aplicação e o cliente retirar 10 também no dispensador de senhas, isso é algo que não poderá acontecer, tudo tem que ser bem pensado e ser bem programado. Trata-se apenas de uma ideia por enquanto.



## **Referências Bibliográficas**

**Abeyasinghe, Samisa** (2008), RESTful PHP Web Services.

**Alves, De Luciano** (2015), Programando passo a passo (Programação básica versão Android Studio).

**Caelum**, Desenvolvimento Web com HTML, CSS e Java script.

**Caelum**, Java e orientação a objetos.

**Carvalho, Kleber Rodrigo de** (2012), Aplicativos Web Pro Android.

**Cookbook, Android** (2011), Android Cookbook.

**Da Silva, Alberto, & Videira, Carlos** (2001), UML, Metodologias e Ferramentas CASE.

**Griffiths, Dawn & Griffiths, David**, Android Development.

**HandBook de TI para concursos**, O Guia Definitivo.

**K19 Treinamentos** (2009) Integração de Sistemas com Webservices, JMS e EJB.

**K19 Treinamentos** (2012), Desenvolvimento Mobile com Android.

**K19 Treinamentos** (2015), Desenvolvimento Web com HTML, CSS e Javascript.

**K19 Treinamentos** (2015), Modelo Relacional e SQL.

**Landis, Joshua & Haff, G Gregory & Weatherford, Zach** (2006), Android Programming Guide for Beginners.

**Lecheta, Ricardo R** (2016) Google Android, Aprenda a criar aplicativos para dispositivos móveis com SDK, 5ª Edição.

**Richardson, Leonard & Ruby, Sam** (2007), RESTful Web Services.

**Lopes, Sérgio** (2013). A Web Mobile, Programe para um mundo de muitos dispositivos.

**Mazza, Luiz** (2012), HTML5 e CSS3, Domine a web do futuro.

**Monteiro, João B.**, Google Android, Crie aplicações para smarphone e tablets.

**Nixon, Robin** (2014) Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition.

**Deitel, Paul & Harvey** (2010) Java como Programar 8ª Edição.

**Deitel, Paul & Harvey** (2016), Android 6 para Programadores - 3ª Edição: Uma Abordagem Baseada em Aplicativos.

**Pereira, Cláudia** (2009), Uma Introdução Às Filas De Espera.

**Phillips, Bill & Hardy, Brian** (2013), Android Programming: The Big Nerd Ranch Guide.

**Silva, Carlos**, Introdução a Programação Android.

**Sklar, David & Trachtenberg, Adam** (2014) PHP Cookbook, Third Edition.

**Smyth, Neil** (2015), Android Studio Development Essentials- Android 6 Edition ©.

**Wiley, John & Sons, Inc.** (2015), Android™ Application Development All-in-One For Dummies®, 2nd Edition.

## Web grafia

**Tutorials point.** Disponível em

<[https://www.tutorialspoint.com/android/android\\_studio.htm](https://www.tutorialspoint.com/android/android_studio.htm)> [Consultado em 21/12/16 às 10:23].

**Como consumir um Web Service pelo Android.** Disponível em

<<http://www.portalandroid.org/comunidade/viewtopic.php?f=119&t=17465>> [Consultado em 06/09/16 às 14:22].

**Javatpoint.** Disponível em <<https://www.javatpoint.com/android-tutorial>>

[Consultado em 17/11/15 às 17:12].

**Receber dados via json em um aplicativo android.** Disponível em

<<http://www.matera.com.br/?s=Receber+dados+via+JSON+em+um+aplicativo+Android>> [Consultado em 17/03/16 às 18:56].

**Como funciona um Webservice REST.** Disponível em

<<http://www.matera.com.br/?s=Receber+dados+via+JSON+em+um+aplicativo+Android>> [Consultado em 17/10/16 às 17:55].

**Chega de segredos com REST e WebServices no Android – Aprenda Fácil.**

Disponível em <<http://www.mobimais.com.br/blog/retrofit-2-consumir-json-no-android/>> [Consultado em 10/09/16 às 15:59].

**Gestão de Filas de espera QSystems.** Disponível em < <http://www.qsystems.pt/>>

[Consultado em 22/04/16 às 12:59].

**Gestão de Filas de espera Qmagine.** Disponível em

<[http://www.qmagine.com/store\\_pt/index\\_2010\\_v2.php](http://www.qmagine.com/store_pt/index_2010_v2.php)> [Consultado em 27/10/15 às 15:00].

**Corporate Inline – Gestão Virtual de Filas de Espera .** Disponível em <[http://sti-consultoria.com/site/pdf/filas\\_espera.pdf](http://sti-consultoria.com/site/pdf/filas_espera.pdf)>

[Consultado em 17/05/17 às 17:00].

**Desenvolvendo e Usando Web Services em Java.** Disponível em <[http://sti-consultoria.com/site/pdf/fias\\_espera.pdf](http://sti-consultoria.com/site/pdf/fias_espera.pdf)> [Consultado em 17/05/17 às 17:00].

**Usando webservices.** Disponível em <<http://www.linhadecodigo.com.br/busca.aspx?w=Desenvolvendo+e+Usando+Web+Services+em+Java>> [Consultado em 17/04/16 às 11:00].

**Introdução ao JSON.** Disponível em <<http://www.json.org/json-pt.html>> [Consultado em 17/03/16 às 20:00].

**Android Developers.** Disponível em <<https://developer.android.com/reference/android/app/Service.html>> [Consultado em 17/01/17 às 11:00].

**Tutorial Web.** Disponível em <https://www.w3schools.com/> [Consultado em 17/06/16 às 19:00].

**Dreamweaver Training and Tutorials.** Disponível em <<https://www.lynda.com/Dreamweaver-training-tutorials/185-0.html>> [Consultado em 17/07/16 às 12:04].

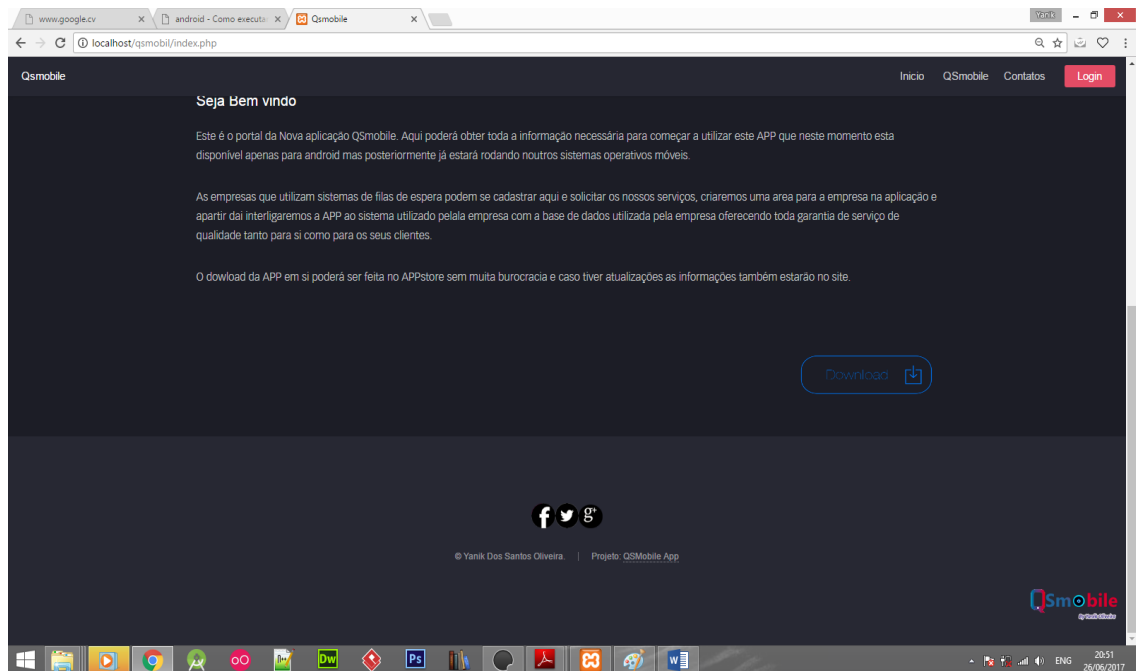
**Android Studio.** Disponível em <<https://developer.android.com/studio/index.html>> [Consultado em 17/08/16 às 22:44].

**Genimotion.** Disponível em <<https://www.genymotion.com/>> [Consultado em 17/08/16 às 22:44].

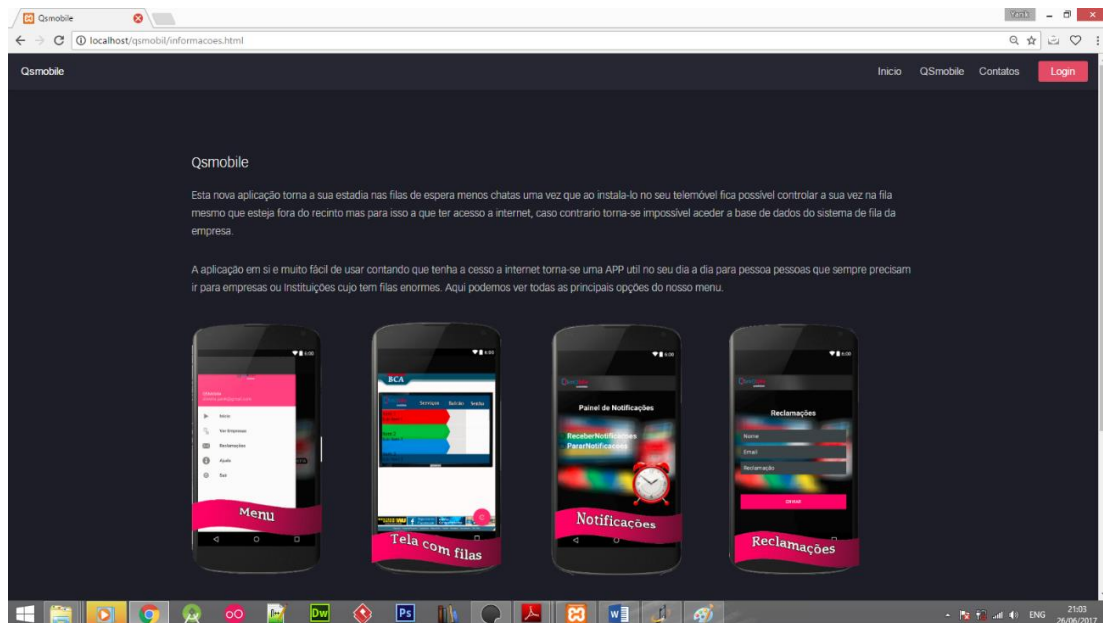
## Anexos

### A) OUTRAS PARTES DA WEBSITE

Logo na parte inicial da página web podemos ver a opção download onde o utilizador poderá baixar a aplicação e instalar no seu dispositivo android ou também poderá baixa-lo diretamente no Google play.



Na tela seguinte podemos ver algumas informações acerca da aplicação, essa parte tem o objetivo de esclarecer ao utilizador qualquer dúvidas acerca das opções do menu da aplicação.

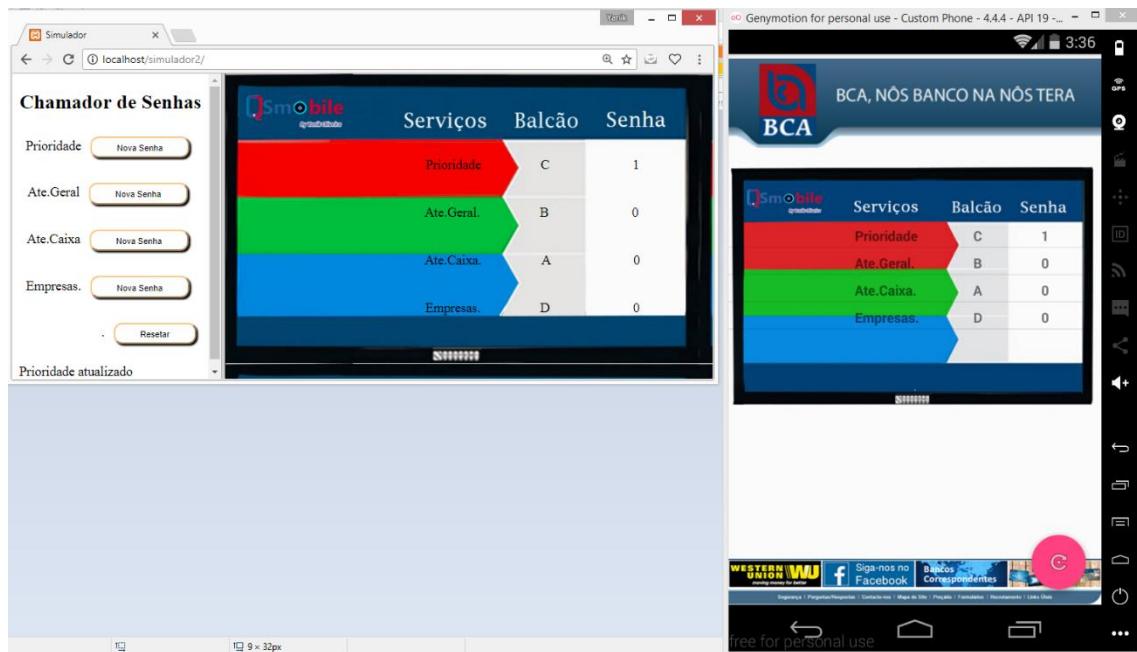


## B) SIMULADOR DE UM SISTEMA DE GESTÃO DE FILAS DE ESPERA

Para que se pudesse ver a aplicação funcionando em pleno mostrando os dados de uma fila, tive que criar um simulador de um sistema de gestão de filas de espera para fazer essa simulação já que no momento não há maneira de fazer. Esse simulador possui um chamador de senhas e um display que mostra o serviço o balcão e senha que esta sendo atendida no momento. Ele foi criado com HTML e PHP e utiliza uma base de dados MYSQL para fazer a manipulação dos dados e mostra-los na tela.



Aqui podemos ver o simulador funcionando juntamente com a aplicação android.



### C) TERMO DE ACEITAÇÃO



**UNIVERSIDADE DO MINDELO**

*Sapientia Omnium Potentior Est*

#### TERMO DE RESPONSABILIDADE de ORIENTAÇÃO

Eu, Paulo Alexandre dos Santos Silva, grau: Mestre,  
declaro que o aluno Yanick Gilson Santos Oliveira, N.º 2858  
Finalista do curso de Informática de Gestão, realizou sob a minha  
orientação o Trabalho de Conclusão Curso/Monografia/Relatório de Estágio/Projeto de Licenciatura  
intitulada: “QSMOBILE: App e Plataforma  
Web para Fila de Espera”.  
e que a mesma foi desenvolvida de acordo com as Normas de Elaboração e Apresentação dos TCC's  
da **UNIVERSIDADE DO MINDELO** e reúne todas as condições para a sua apresentação e defesa.

Mindelo, 7 de Julho de 2017

O Orientador

P. A. dos Santos Silva